softalk for the IBM Personal Computer April 1984

April 1984 Volume 2 $2.50

softalk

for the IBM Personal Computer

MUCH MICE!

ANNOUNCING:
- **PORTABLE PC**
- **SYMPHONY**
- **MULTI-USER dBASE II**

FOUR LOGOS FOR THE PC

RESULTS:

THE MOST POPULAR SOFTWARE POLL

It Reads, Writes and It Paints in 3-D, Keeps and Talks to

It's called "OPEN ACCESS," and it's the result of 60 man-years of effort to create a truly do-it-all, super-program—one that can perform virtually every task you're ever likely to encounter.

The beauty of it is, all that capability resides on a single program. You don't have to re-enter data. Or spend time trying to get unmatched programs to work together.

OPEN ACCESS takes its name from the source of its power—a relational data-base manager that gives you access to more data in more ways than any comparable software.

OPEN ACCESS includes an electronic spreadsheet, 3-D graphics, word processor, appointment scheduler and telecommunications module—all revolving around the powerful information manager.

See us at COMDEX, Booth W648

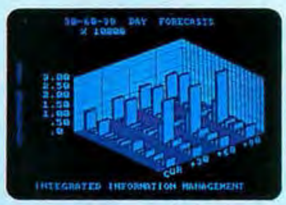
1

INFORMATION MANAGEMENT—THE MASTERMIND. This advanced data-base manager stores and retrieves multiple files quickly, easily and reliably. What's more, it shares all information with the other programs, so you never have to re-enter the same data twice.

2

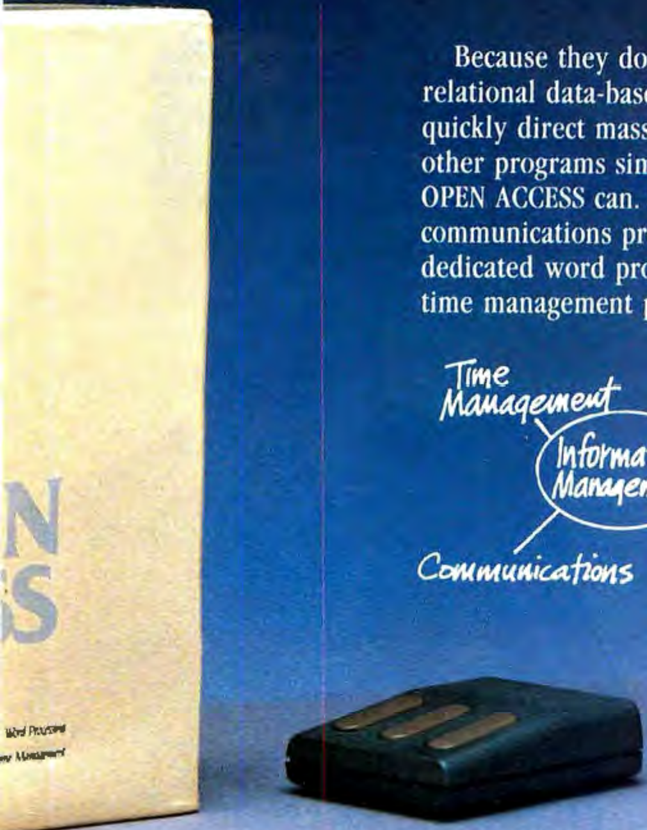
ELECTRONIC SPREADSHEET—NUMBER CRUNCHING AND GOAL SEEKING. It helps you produce forecasts, cost estimates and "break-even" points—in seconds, instead of hours or days. Best of all, it allows "goal seeking." Ask, for example, "What sales must I have the rest of the year to net $1 million?," and OPEN ACCESS will figure it out!

3



3-D GRAPHICS—NOT JUST PRETTY PICTURES. These graphics distill raw data into trends that can be instantly visualized, helping you discern the important facts from a wealth of information.

does Arithmetic. Your Appointments the World.



Because they do not have a dedicated relational data-base manager that can quickly direct massive amounts of data, other programs simply can't do what OPEN ACCESS can. Some don't have a communications program, others no dedicated word processor. None have a time management program.

There's just one conclusion: At $595,* OPEN ACCESS can do more for you than any other comparable business program on the market. Bar none. But the only way for you to be convinced is for you to see OPEN ACCESS work its magic on your work load. So call your local software dealer today, or call us at SPI, at 619-450-1526.

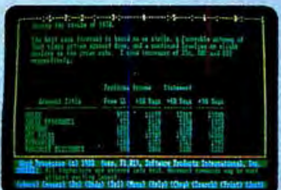
*Introductory price

SPI

SOFTWARE PRODUCTS INTERNATIONAL

10240 Sorrento Valley Road
San Diego, CA 92121

4



WORD PROCESSING—EDITOR EXTRAORDINAIRE! Superior word processors make it easy to correct typos, change words, shuffle paragraphs and format documents. This is one of that breed. Use it to write efficient memos, letters, proposals and reports.

5

TELECOMMUNICATIONS—YOUR LINK WITH THE WORLD. This program gives you access to virtually any other computer system in the world. Not only can you transmit and receive reports from your colleagues, you can also subscribe to special data banks that know everything from GM's stock price to the relative humidity in Genoa. Now that's power!

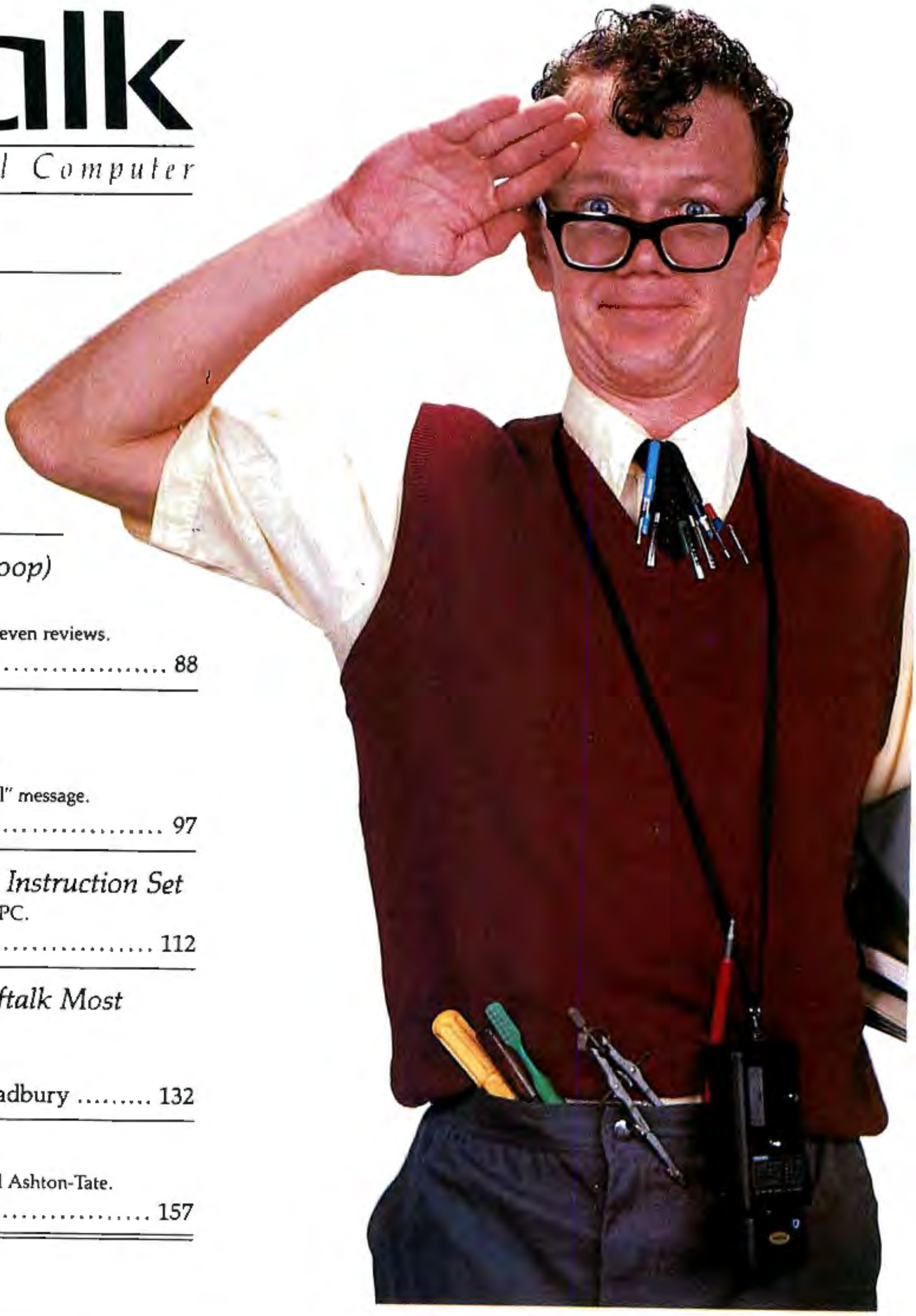
6

TIME MANAGEMENT—CONSERVING YOUR MOST PRECIOUS RESOURCE. This module helps you keep track of all your appointments, hour by hour, day in and day out. It alerts you to standing obligations, automatically coordinates meeting times with other busy professionals, and lists all your associates on a Rolodex™-like file.

softalk

for the IBM Personal Computer

Features



Columns

Departments

Cover photography by Kevin McKeon and Tim Durr. Special thanks to Woody for his catalogue of poses.

Index to Advertisers

Composition by Type Works, Pasadena, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.

ISSN: 0733-2173. *Softalk for the IBM Personal Computer* is published monthly by Softalk Publishing Inc., 7250 Laurel Canyon Boulevard, North Hollywood, CA 91605; telephone (818) 980-5074. Second-class postage paid at North Hollywood, California, and additional mailing offices.

Postmaster: Send address changes to Softalk/IBM, Box 7040, North Hollywood, CA 91605.

Subscriptions: Complimentary six-month trial subscription to all owners of IBM Personal Computers, Compaq, or Corona computers in the USA and Canada. If you own a PC, Compaq, or Corona but you aren't receiving *Softalk for the IBM Personal Computer*, send your serial number and mailing address to Softalk/IBM Circulation, Box 7040, North Hollywood, CA 91605. Non-PC-owner subscriptions: $24 per year. Please allow six to eight weeks for processing. *Softalk for the IBM Personal Computer* is totally independent of International Business Machines.

Back issues (from June 1982): $3.

Problems? If you haven't received your *Softalk* by the fifteenth of the month, or if you have other problems with your subscription, Marsha Stewart can help out. Call (818) 980-5074 or (800) 821-6231.

Moving? Send new address and a recent mailing label from your old address to Softalk/IBM Circulation, Box 7040, North Hollywood, CA 91605; telephone (818) 980-5074. Please allow six to eight weeks for processing.

THINK

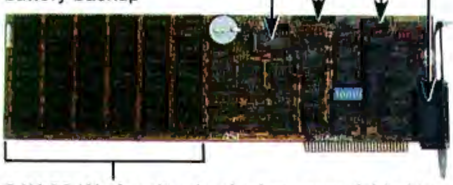
Albert Einstein did it. . . . Auguste Rodin sculpted it . . . and Thomas J. Watson, Sr., put it best. But we found eleven other ways to say it, and we mixed them all up in the boxes. Sort out as many of them as you can and send the results to Tangletalk, Box 7040, North Hollywood, CA 91605. The scholar who comes up with the closest to a complete list will win $100 of the software of his or her choice from among the products advertised in this issue. If more than one entrant manages to untangle all eleven words, we'll rely on our random number generator to select a winner.

Number One Add-On Products for IBM PC

Getting the most out of your personal computer.

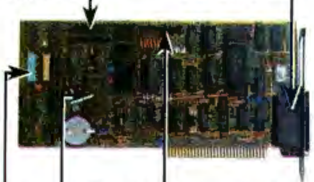
AST Research Number One Add-On Products let you realize the full potential of your IBM PC or PC-XT without wasting valuable slot space. You can take advantage of more of the capabilities IBM designed into the PC while leaving space for future enhancements as they are introduced, by combining your memory and input/ouput requirements on a single card!

SixPakPlus™:



64K-384K of parity checked memory. Added to a PC or XT with a fully populated 256K system board, the SixPakPlus can bring the system memory to 640K, the maximum addressable user memory.

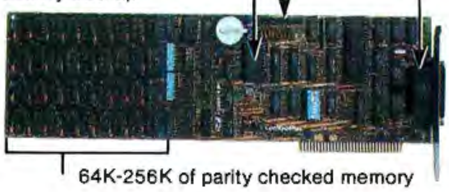
I/O Plus II™:



MegaPlus II™:

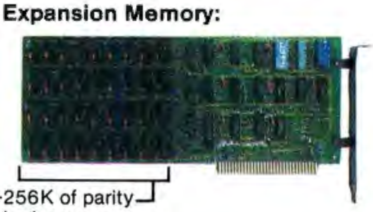
64K-512K of parity checked memory. The basic card expands to 256K, and with the MegaPak extension expands to an additional 128K or 256K of parity checked memory.

ComboPlus™:



64K-256K of parity checked memory

MP Expansion Memory:

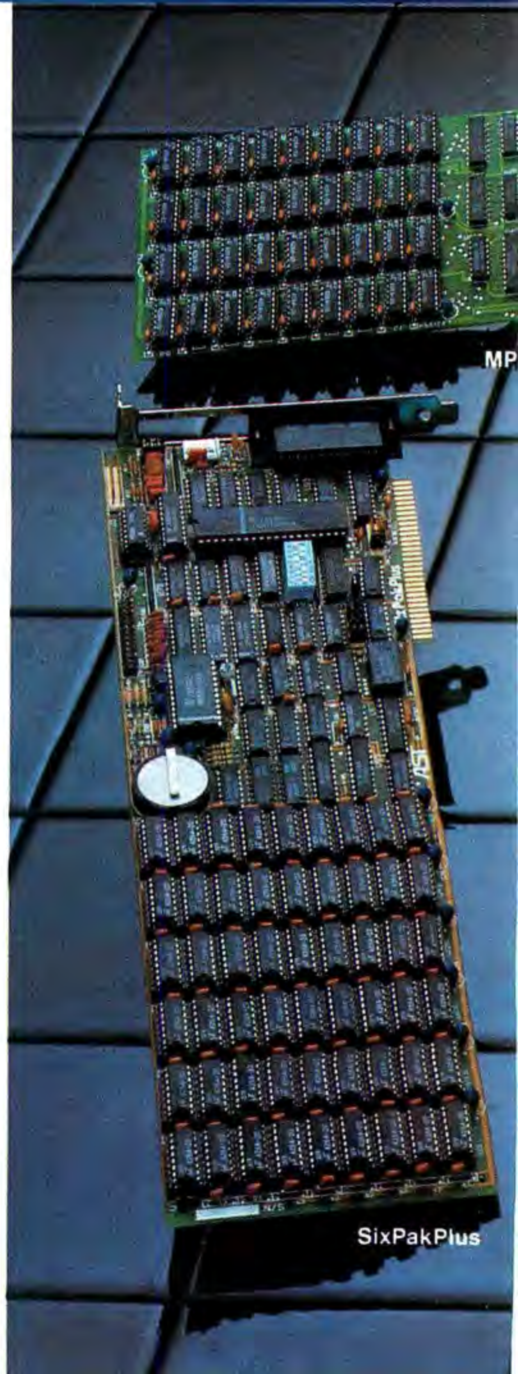


64K-256K of parity checked memory

AST-PCnet™:

- CSMA/CD 1 Mbps baseband Local Area Network
- Uses standard 75-ohm CATV coaxial cable capable of running up to 7,000 feet
- Interconnects multiple PC's (57,000 addressable limit)
- All PC-compatible disk drives and printers are shareable
- Networked access to mainframes via shared AST-3780
- Users can execute commands remotely on shared PC
- File lock-out
- DOS 1.1 and DOS 2.0 compatible

Ask for AST Research Number One Add-On Products, available at Computerland, Entré, Businessland and other computer stores worldwide. For the dealer nearest you, or for complete product information contact AST Research, Inc., (714) 540-1333/863-1333 TLX 753699 ASTR UR



User Memory from 64K-512K — When added to your existing system memory, brings your PC up to its maximum of 640K. You can run larger spreadsheets or create larger in-memory databases. You can also use the additional memory along with the supplied AST SuperPak™ software which includes SuperDrive™ and SuperSpool™.

Serial Ports — Using a modem your PC can communicate with other computers over telephone lines. By connecting a serial printer, you can obtain high quality print output. Other serial devices such as a plotter or mouse may also be connected to a serial port.

Parallel Ports — Used for connecting a parallel printer to your PC for high-speed, draft quality printouts.

Clock Calendar — With the on-board battery, the clock-calendar feature automatically maintains the correct date and time, even when the PC is turned off.

Game Adapter — Allows you to connect an IBM-type joystick to your PC so you can play the multitude of arcade quality games. Or you can use the joystick input for other applications by writing your own programs.

Other Communications Products — AST Research also extends the capabilities of your IBM PC with mainframe communication products including 3270 SNA and 5251 terminal emulation, 3780 RJE support, and AST-PCnet™ — the Local Area Network designed for the IBM PC.

AST *Quality*

All AST Research enhancement products come with the **AST "Plus"** — our unsurpassed reputation for quality, reliability, after-the-sale support, and overall design excellence — which give our products the best price/performance ratio in the industry!

PCnet is a registered trademark of Orchid Technology, Inc.

crosstalk

Public Domain C

In the August 1983 *Softalk* Peter Norton introduced us to the Microsoft C compiler. I have since purchased this compiler and have been busily learning the C language and trying to write statistically oriented application programs. As veteran C programmers could have warned me, C does not compare to Fortran when it comes to number crunching. Undaunted, I have started to develop descriptive statistical library routines.

It was while laboriously trying to optomize these routines that it occurred to me that I was reinventing the wheel. A very large foundation of Unix-based public-domain C programs must already exist. Bell Laboratories and universities such as the University of California, Berkeley, must have developed hundreds of useful C utilities and library routines that are available free or for nominal fees. There must also be user groups dedicated to implementation of C on the PC that I might join to contribute my programs and share with others. I for one would appreciate any help you or fellow readers can offer introducing us to these sources.

Alan Hoshor, Escondido, CA

Here's some information: Try Digital Equipment Corp. (DEC), which publishes a Unix software handbook that includes public-domain program descriptions. Call your local DEC dealer for a copy. Also, DECUS, DEC's worldwide user group, has at least one nine-track magnetic tape full of public-domain C routines. You must be a member to order one, but membership is free. Contact DECUS, MR2-3/E55, One Iron Way, Marlborough, MA 01752 (617-467-4167). \usr\group provides a forum for Unix/C info exchange. Contact \usr\group, Box 8570, Stanford, CA 94305-0221 (408-986-8840). USENIX Association also provides C information. Contact USENIX Association, Box 7, El Cerrito, CA, 94530 (415-528-UNIX). Bulletin boards with information: Capital-PC C/Unix BBS (703-321-7003); 300/1200 baud. Capital-PC Statistics BBS (301-596-3596); 300/1200 baud.—Rex Jaeschke.

Falling into a Chasm

In Marc Melcher's December 1983 letter to "Crosstalk," my user-supported program *Chasm* was identified as coming "from Freeware." Although *freeware* has generally been adopted by users as a generic name for a form of noncommercial program distribution, "Freeware" is in fact a trademark of a specific company, The Headlands Press. Since *Chasm* is not a product of The Headlands Press, use of the word *freeware* is both inappropriate and misleading. Mr. Melcher's confusion is undoubtedly the result of the similarity in the way *Chasm* and The Headlands Press's software are distributed.

Anyone wishing to try a free demonstration version of *Chasm* is invited to send a formatted disk and a stamped return mailer to me at 136 Wellington Terrace, Lansdale, PA 19446. A $30 payment is suggested from those who find the package useful. Those who make the payment become eligible to receive a free upgrade to a compiled version of the program that runs substantially faster and incorporates additional features.

David Whitman, Hanover, NH

Runaway Metaphor

I must correct a metaphor Ken Landis used in his December "Micro Finance." Mr. Landis referred to a "runaway nuclear reactor." All the power reactors in the United States are inherently incapable of becoming runaways. This is not due to a design feature devised by a human and fallible engineer or the quality or training of the operators, but to an inherent rule of physics that is as immutable as the law of gravity. A runaway train or a runaway truck or runaway inflation, perhaps, but not a runaway nuclear reactor.

Willis G. Frick, Rosemead, CA

On Errors from Typesetting Programs

February's "Crosstalk" contained a letter ("Typos in Programs") about a subject close to my heart: the typesetting of computer programs. The writer is quite correct in complaining about your practice of typesetting listings. Not only does this invite errors, but it makes it extremely difficult for the reader to tell where spaces should be inserted. In this regard, I'm afraid that *Byte* is way ahead of you.

By the way, in spite of your response to this reader's letter that you typeset directly from disk files, I find the following code on page 14:

```
90 IF MIN% = 59 AND SEC% - 59
THEN 10
```

Obviously, it should read *Sec% = 59*. I realize that this isn't from an author, but it illustrates the problems you introduce by typesetting! Was the original sent as a listing? You should've left it alone!

Generally, however, you do a good job. Keep it up.

Dave Burrows, State College, PA

Toolbox Typo

John Socha's article, "Behind the Scenes...The Making of a Toolbox Program" (February

Compare Best's Professional Finance Program™ to anyone's. The competition just isn't up to par.

Use this score card to compare Best's Professional Finance Program to the Home Accountant Plus, or any other financial program. You won't find any that can make financial management simpler or more effective.

The Professional Finance Program is software that has all the features you want, but is simple to operate. The keys to its simplicity include a help menu linked to current activity, and other aids like budget, account, and tax/sort names displayed on the screen. The program also comes with a fully indexed user's manual and is backed up by Best's free customer support.

The Professional Finance Program is also a great help at tax time. It provides financial information for easy input into Best Program's PC/TaxCut™, a tax preparation and planning package. Combine the two programs and you have a complete financial and tax package.

If you already own the Home Accountant Plus™, don't consider that a handicap, because Best is offering you a $100 rebate. The program is compatible with the IBM PC, PC/XT, COMPAQ, Columbia Data Products, and Eagle Computers. For more information about the rebate or the Professional Finance Program, call 1-800-368-2405. In Virginia call 1-703-931-1300. Or write to Best Programs, 5134 Leesburg Pike, Alexandria, VA 22302.

Score Card

Feature	Professional Finance Program™	Home Accountant Plus™*	Other
Budget Codes	1,170	200	
Tax ID Codes	99	1	
Transactions Displayed Per Screen	16	1	
Number of Checkbooks	26	5	
80-Column Screen	Yes	No	
On-Screen Calculator	Yes	No	
Password Protection	Yes	No	
Address Book for Payees	Yes	No	
Speedy Compiled Basic	Yes	No	
Full Use of IBM PC Function Keys	Yes	No	

*Home Accountant Plus™ is a trademark of Continental Software.

BEST PROGRAMS

"The Quality Software Company"

1984) serves several useful purposes. First, it provides a utility program that will clear the screen from DOS. Second, it gives an interesting example of the use of check digits. Finally, and most important for followers of Mr. Socha's articles, it allows readers to re-create utility programs in future columns by simply typing in the data statements. Mr. Socha is to be congratulated for a fine article and an excellent idea.

When I typed in the data in exhibit 4, merged this file with the Writecom file, and ran the program, I got messages saying, "Line 1020 may be bad" and that I was "Out of data at line 280". Apparently the program expects twenty-five pieces of data (line 1000) but lines 1010–1040 give only twenty-four pieces. This accounts for the "Out of data" error when the program attempts to read the twenty-fifth number, and if this number belongs on line 1020, this would also explain why the XOR product of line 1020 doesn't equal its check digit.

The XOR product on line 1020 is 175. The program expects it to be 191 and therefore says line 1020 may be wrong. It can be seen (by actually comparing the bit patterns of 175 and 191, or by taking 175 XOR 191) that 16 should be added to line 1020 to make the XOR product equal the check digit. By trial and error, I found that placing the 16 between the 205 and 51 works. Once I did this, when I entered DOS and gave the command *cls*, the screen cleared and the A> prompt was returned.

John D. Coffin, Howell, NJ

What Else? An Otrona Attache

As a fan of the Otrona Attache portable computer, I was amused by your parenthetical comment, "The new version (Otrona Attache) has sprouted (what else?) an 8088 for PC compatibility," in January's "Comdex: The *Rite* of Autumn." What else, you ask? Otrona's 8:16 actually contains an 8086 and 256K standard memory for PC compatibility, along with a Z-80A and 64K for CP/M operation.

Also standard are two 360K 5 1/4-inch floppy drives, two RS-232 ports, a real-time clock, programmable three-channel sound synthesizer, CP/M 2.2 and DOS 2.1 operating systems, and Otrona utility programs for graphics, RPN calculator, message and alarm clock, screen dump, and disk management.

You were correct in noting that Attache is the smallest and the lightest of the compatibles at eighteen pounds. It's also the fastest, but, as you say, $3,795 isn't cheap. However, what do other so-called PC clones (portable or desktop) cost with a similar configuration? What does a loaded PC cost? How about a review of the Otrona Attache?

Mike Hutmacher, Berthoud, CO

The Tale of I/O for Standard Devices

Your fine publication had one inaccuracy in the February "System Notebook." Mr. Boyd evidenced a deficiency in his understanding of redirection of I/O for standard devices. The *print* command and DOS behaved as they should when DOS redirected its screen-displayed info to a disk file instead of to the printer. You see, the printer is not the standard output device—instead DOS considers the video display to be the standard output device.

Mr. Duncan might like to know how one person is using his CLEAN utility. I enter my assembler code using the *WordStar* document mode (I despise tab characters and prefer spaces). I then use his utility to create a file acceptable to the assembler.

Rick French, Campbellsville, KY

A Positive Mail-Order Experience

After reading about mail-order software problems so many times, I wanted to share my good experience with your readers. I purchased *BASIC Aids*, after reading about it in your classified section, to help me develop BASIC programs for my employer. After I'd used the product a short time, my employer decided

An Industry First in Word Processing Software:

WordPlus-PC™ featuring the BOSS™. Word processing so smart it can even spell 100,000* words.

Meet the BOSS. WordPlus-PC's incredible new Built-in On-line Spelling System.

WORD PROCESSING AND SPELLING COMBINED:

WordPlus-PC featuring the "BOSS" is a whole new technology in word processing software. Finally, the most powerful word processing capabilities have been combined with a built-in electronic spelling system which not only *checks your spelling* but also *corrects your spelling.*

The "BOSS", an acronym for Built-in On-line Spelling System, eliminates embarrassing spelling errors in your letters and stops time consuming trips to the dictionary.

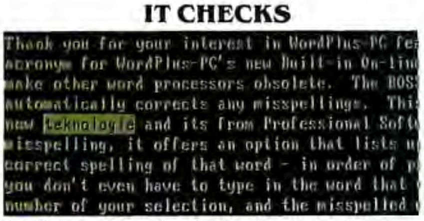
But WordPlus-PC featuring The "BOSS" will do far more than just check and correct your spelling. It's been designed to be exceptionally easy to use and yet contain all the powerful features you expect and more. Like built-in mail merge for personalized form letters, invoice generation and the ability to merge information created by most other popular programs such as 1-2-3™ and dBASE II.™ And if you have a question, just press the HELP key to get back on track.

You can also print bar graphs and other charts,** easily move columns, scroll horizontally, execute global search and replace, boilerplate text, and even print proportionally spaced on selected printers. And WordPlus-PC is compatible with virtually all popular letter-quality and dot matrix printers.

*With 90,000+ word standard dictionary and ability for a user to add over 10,000 "custom" words.

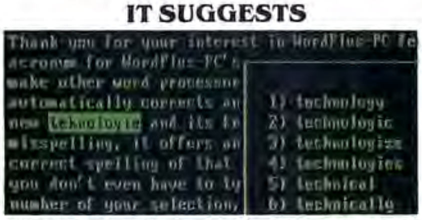
HERE'S HOW THE BOSS WORKS FOR YOU:

IT CHECKS



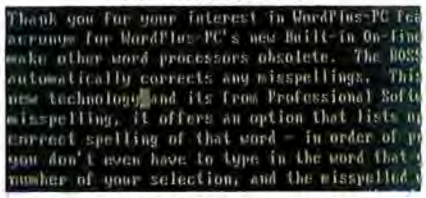
At the touch of a button, virtually anytime during typing or after the creation of a document, The "BOSS" locates and *highlights misspelled words* on your screen. The "BOSS" can even check the spelling of a word directly after it's been typed in.

IT SUGGESTS



When a word is misspelled, the user can ask The "BOSS" for suggestions as to how to correctly spell the word. With only one keystroke, The "BOSS" will display, in a dynamic on-screen window, up to eight *spelling suggestions in the order of probable phonetic correctness.*

IT CORRECTS



And The "BOSS" *Auto Correct* feature enables users to "fix" these misspelled words directly in text with a single keystroke. The "BOSS" is a total spelling system that Checks, Suggests, and Corrects your Spelling. All built-in.

THE TOTAL WORD PROCESSING SOLUTION — PLUS ALL THE HELP YOU'LL NEED.

With over 70,000 word processors in use worldwide, Professional Software, Inc. has designed WordPlus-PC to contain all the powerful features you expect — and to be the easiest word processor to use anywhere. But there will always be questions. That's why Professional Software has an entire bank of Nation-wide TOLL FREE customer HELP lines to support our registered users. This service is available at No Charge during the 90 days following the receipt of your registered users card.

Demand to see the BOSS in person
Visit your local professional computer dealer or call us toll free for the dealer nearest you. Once you witness the BOSS in action, you'll see that all other word processors have become totally obsolete.

Demand to see the "BOSS" today! Call us toll free, 1-800-343-4074.

Professional Software, Inc.

51 Fremont Street, Needham, MA 02194 Telephone (617) 444-5224 Telex 951579
Now Available for IBM-PC and Compatibles, DEC Rainbow, TI Professional, and Victor 9000.
Dealer, distributor, OEM manufacturer, and international inquiries are invited.

WordPlus-PC and The "BOSS" are trademarks of Professional Software, Inc.
1-2-3 is a trademark of Lotus Development Corp., dBASE II is a trademark of Ashton-Tate.
Specifications are subject to change without notice.

WordPlus-PC was designed and written by Andres Escallon.
**With IBM dot matrix and Diablo 630 ECS printer, bar graphs and other charts can be printed inside text.

What should you look for in personal

Before you go looking for personal computer software, you should know what personal computer software looks like.

(You're not alone if you don't already know that software programs come in a package.)

Programs are "pre-recorded" on cartridges, tapes or diskettes. And, although you can't tell by looking at these cartridges, tapes or diskettes, the programs on them can be very different.

What you put in is what you get out.

What happens when you play a high-quality tape on a high-quality recorder? Superior sound.

This analogy can also hold true with software. The better the program quality, the better the result —be it improved productivity or creativity.

IBM Personal Computer Software is both tested and approved by IBM. And these programs are designed to take advantage of an IBM personal computer's many advanced hardware features.

What the value is.

What are improved math skills worth? More efficient inventory control? Faster communications? What is accomplishment worth?

Any way you look at it, the value of personal computer software is the value of doing your best.

What compatibility means.

Many of the same software programs that run on the IBM Personal Computer you use at work will run on other IBM personal computers you use in other places. So you can, for example, continue in your family room what you started in the classroom or boardroom. (Or the other way around.)

The Little Tramp character licensed by Bubbles, Inc., S.A.

to switch to COBOL. Even though *BASIC Aids* is a good product with very helpful documentation and debugging, I no longer had a need for BASIC tools. I wrote to the Tulsa Computer Consortium explaining my situation and requested a refund. My money was promptly and cheerfully returned. I realize that very few software houses have a friendly refund policy and that my experience probably is unique.

Donnie W. Tenney, Williamston, NC

Hail, Columbia

I've been the proud owner of a Columbia PC compatible for six months now. Please allow me to speculate why your magazine rarely prints an article about and/or including the Columbia family of PC compatibles.

Is it because they don't contribute their hardware or software for review? Maybe because your "generous advertisers" might get upset? Or don't you print articles on the Columbia simply because no other manufacturer can offer a complete compatible computer including fifteen hundred dollars worth of free software for their prices? Could it be you feel that it's immoral to underprice all the leading manufacturers and it could mean economic disaster for many of your acquaintances?

Whatever your reasoning, don't you think it's time for the public to know there's a compatible out there for half the cost with some very good software? I always thought the press prints the truth even if it's rather unpleasant to the majority of its contributors and advertisers. C'mon guys, let's get with it.

By the way, this expresses my feelings conclusively. I hardly hear anything about my Columbia, and since it's totally compatible I can only guess it's ignored from a combination of the motives I mentioned. I certainly hope you have the integrity to print this.

Fred Tabaracci, New York, NY

The Much-Documented *Shell* Command

Dan Rollins's article, "New BASIC Commands Uncovered" (January) was most useful, especially his explanation of how to avoid a crash when returning to BASIC after a *shell* command. Since I usually call DOS from BASIC in direct mode rather than from a running BASIC program, I have created two strings that I assigned to keys (using *Prokey*), one to go to DOS and the other to return to BASIC.

To go to DOS from BASIC:

```
PPPP=PEEK(48):PPPPP=PEEK(49)<enter>
SHELL<enter>
BREAK ON<enter>
```

The variable names are purposely long so as not to conflict with variables in a program. I turn break on routinely when going from BASIC to DOS because BASIC seems to destroy the *break on* command.

To return from DOS to BASIC:

```
EXIT<enter>CLS<enter>POKE
48,PPPP:POKE
49,PPPP<enter>
```

The *cls* command clears the screen of garbage left over from the DOS activity. The garbage might otherwise produce a syntax error if it's on the same line as the two *poke* commands.

John R. Herzfeld, Mercer Island, WA ▲

CONTEST WINNERS

If you chewed your fingernails down trying to sort out the mixed-up PCs from January's "MicroCommotion" contest, don't feel bad; you had plenty of company. The contest was one of our most popular ever, and the hundreds of eager puzzle solvers fell into several distinct groups: correct, almost correct, and off-in-left-field.

Those of you in the last category just plain got it wrong. Many of you clearly believe that Godzilla was from Down Under, drives a polka-dotted hot rod, and uses *Styx* to keep track of his tax records. You may have a future in politics.

The almost-corrects comprised the sizable number of you who took the moniker "Logical Moves" seriously. True, logic was required to reach the correct solution, but as many of you pointed out, logic alone only narrowed down the possibilities to two. In the end, intuition was required to determine that George's PC was in East Cupcake, Texas—so far as we know, there was no definite clue that such was the case. For many of you, that intuitive leap was fatal.

Of course, that doesn't explain how H.W. Neff of San Leandro, California, managed to write a computer program that allowed his PC to reach the correct solution. "Even the infamous random number generator should be impressed," wrote Neff. Well, sorry, but the random number generator objects to the label "infamous" on the grounds that it casts aspersions on the objectivity of that august arbiter.

As a result, we're happy to report that of the 119 humans and 1 computer that got the right answer, it was Jerold S. Malina, from Morton Grove, Illinois, a member of the former group, who received the nod from RND.

For those of you who were frustrated by the slight illogicality of "MicroCommotion," we solemnly promise that our next two contests, not to mention last month's, will have but one right answer. Happy deducting. ▲

Incredible Trainers!

No matter what level IBM Personal Computer training you want, our interactive tutorial software gives you the answers. Easier, faster, smarter.

Take **The INSTRUCTOR,** for first-time users. PC Magazine called it, **"The best . . . introduction to the PC around."** It's #1 because it takes the work out of learning initial PC operations.

And **Now! Brand New!** Setting a new standard for interactive training, we give you . . .

PROFESSOR DOS.

You've got to see (and hear) it to believe it! Designed for PC users ready to learn higher level PC concepts, including DOS commands. Professor DOS will challenge you, amuse you, and amaze you.

Both programs use vivid graphic images, creative animation, sound and living color to take you from mystery to mastery of your IBM PC. Each is totally self-paced. And they're "people-literate." So you really do learn.

The most sought-after computer-assisted training anywhere. Available at IBM PC dealers and software retailers everywhere, separately or in a tutorial set. Affordable at a suggested list price of **ONLY . . .**

$44.95 — The INSTRUCTOR
$59.95 — Professor DOS
or
$94.95 — Tutorial Set

We've made **The INSTRUCTOR** and **Professor DOS** so realistic, you'll agree . . . **THEY'RE INCREDIBLE!**

individual
SOFTWARE INCORPORATED

24 Spinnaker Place
Redwood City, CA 94065
(415) 591-4166

Requires MS-DOS, any IBM Personal Computer or Compaq Personal Computer with at least one diskette drive and a monochrome or color display.

Phone and dealer inquiries welcome.

QUESTIONS & ANSWERS

by Nancy Andrews

Q: I'm trying to compile a fairly large BASIC program using the IBM BASIC Compiler version 1.0 running under DOS 1.1, and I've apparently run out of working storage in the compiler. The program is about forty-two thousand bytes of BASIC code, and I'm using the /o, /s, /w, and /x options of the compiler. The compilation gets to within about thirty statements of the end and then begins to give a series of "TC" (Too Complicated) messages—then, finally, the message "Memory overflow" and the cryptic note "22151 Bytes Available, 0 Bytes Free".

I have 320K in my system, but the compiler seems to be unable to take advantage of it, since the same abort occurs on systems with larger or smaller memories. Is there any way I can increase the compiler's working storage? Is a later version of the compiler available that will handle larger programs? Is there any way I can compile the program in pieces and link them together? The manual gives little or no indication of what the limitation is or what to do when it is exceeded. Any suggestions would be appreciated.

Alan W. Jones

A: There's really not much that can be done. There is no way to increase working storage for the compiler, nor is there a way for you to compile separate modules and link them. Your best bet is to look at the *chain* command. If several parts of your program are well defined, perhaps you could use *chain* to swap them back and forth.

Q: I have a PC and use DOS 1.1 and BASICA. I am writing a menu-driven utilities program and would like to know how to accomplish two things from BASICA:

1. Have the utilities program make a hard copy listing of some other BASICA program and then continue its execution. If possible, I would like to have the listing made of non-ASCII–saved programs.

2. Execute a DOS command from BASICA—such as *diskcopy, format*, or *chkdsk*.

Can these things be done with DOS 1.1?

Jerald Watts, Jr.

A: It isn't possible to make a hard copy listing of a non-ASCII BASICA program from within BASICA. But if you save your BASICA program as an ASCII file, you can just open it for input and send it to the printer character by character.

The BASIC *shell* command allows you to execute DOS commands from BASIC or BASICA—but it only works in the 2.0 revision of BASIC and DOS. See the article "Undocumented Commands in BASIC 2.0," by Dan Rollins, in the January issue.

Q: I was extremely pleased to see that IBM's BASIC version 2.0 has the control-print-screen option to cause text sent to the screen to be sent to the system printer as well. However, I was disappointed to find that this option isn't available on what I believe to be IBM's latest BASIC compiler—version 1.0.

Does IBM have a BASIC compiler that acknowledges the control-print-screen option? If not, is there any way of accessing this facility when using compiled programs?

Paul R. Howarth

A: What you can do to get the same effect is redirect output to the printer using the DOS 2.0 I/O redirection capability. To do this, simply type the following from DOS:

```
BASICA filename >PRN
```

and output will be sent to the printer as well as the screen.

There is also a way of accessing the shift-print-screen option from within a BASIC program. Here's a short subroutine you can include in your BASIC programs. It calls Interrupt 5, which is the screen-dump routine. Each time you need it, just put a *call* statement in your program.

```
10  READ X$
20  PRT$ = ""
30  WHILE X$ <> "999"
40    PRT$=PRT$+CHR$(VAL("&H"+X$))
50    READ X$
60  WEND
70  DATA CD,05
80  DATA CB
90  DATA 999
100 PRT = PEEK(VARPTR(PRT$) + 1) + 256 *
    PEEK(VARPTR(PRT$) + 2)
200 REM Whenever you need to print the screen, include the
    following line:
300  CALL PRT
```

Q: I am having a problem compiling a BASIC file. I have IBM's compiler—complete with the Easy.bat file, something I desperately need, since I don't understand much about compiling, linking, and such. This is what I get when I enter *easy filename*:

```
Binary Source File
22151 Bytes Available
22151 Bytes Free
0 Warning Error(s)
1 Severe Error(s)
Fatal Error:  Invalid Object Module
Input File: <filename> pos: 00FA5 Record Type: 44
```

The file I'm trying to compile is 19,968 bytes long. When I compare the file to a copy, I get the message: "EOF mark not found, Files compare ok". Is this the severe error? If so, how do I correct it?

Ben Lee

A: There are two things for you to look into in order to solve your compiler problem. First, be sure you have saved your BASIC program in ASCII. To do this, type *save "filename"*,*A*, and then try to recompile. If that doesn't help, look at the .lst listing the compiler produces. This listing includes all error messages the compiler generates. To see what's in this list, enter the following from DOS:

```
TYPE filename.lst
```

You should be able to locate the error and then use the manual to find the correct syntax. It's possible to have a program that runs correctly in interpreted BASIC but that generates a syntax error in compiled BASIC.

Q: I am an experienced programmer but not at the assembly language level on the IBM PC. Recently, I wrote a program for a friend in *dBase II*. A short time after he began using it in his business he asked me if I could password-protect entry into the program because the data were sensitive. After some research in the DOS 2.0 manual I discovered there are certain new escape sequences that can be used to control output to the screen. Having programmed on other micros, I was familiar with the use of control sequences. I thought if I used the CHR$(27) feature of *dBase* with the other necessary characters, I could turn off the echo while the operator typed the password, then turn it back on. Of course, it didn't work.

After some more reading I was sure I had found the answer by in-

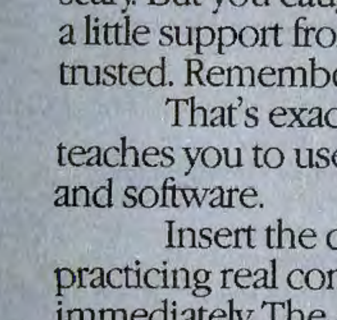
Learn to use your computer the way you learned to ride a bike.

Your first time on a bike was scary. But you caught on fast, with a little support from someone you trusted. Remember?

That's exactly the way ATI teaches you to use your computer and software.

Insert the disk, and you're practicing real computer skills immediately. The ATI program guides you, step by step, simulating the actual program.

Our method is faster because you learn by doing, interacting with the screen. Our unique split screen approach lets you learn in half the time of any other training program. That's why we're the choice for many Fortune 500 companies and have sold over 100,000 programs to date.



Also, ATI has training disks and handbooks for most of the popular software programs.

Before you know it, using your IBM PC will be as natural as riding a bike. Ask for ATI at your computer store. Or, use this coupon to order by mail.

Please rush me ATI Training Power™ programs, at $75 each for this software:

Integrated Software
- ☐ Lotus 1-2-3
- ☐ SuperCalc³

Database Management
- ☐ dBASE II
- ☐ EasyFiler

Word Processing
- ☐ WordStar
- ☐ EasyWriter II
- ☐ Perfect Writer
- ☐ Spellbinder
- ☐ Microsoft Word
- ☐ Multi Mate

Financial Planning
- ☐ Multiplan
- ☐ SuperCalc
- ☐ VisiCalc
- ☐ Perfect Calc
- ☐ EasyPlanner
- ☐ Microplan
- ☐ TK! Solver
- ☐ PeachCalc

Programming
- ☐ BASIC
- ☐ APPLESOFT BASIC

Accounting
- ☐ BPI Gen. Acct.
- ☐ Peachtree General Ledger*
- ☐ Peachtree Accounts Receivable*
- ☐ Peachtree Accounts Payable*

*PC (MS) DOS formats only

Operating Systems
- ☐ PC DOS
- ☐ MS DOS
- ☐ CP/M

How to Use
- ☐ How to Use Your IBM PC
- ☐ How to Use Your COMPAQ
- ☐ How to Use PC jr.
- ☐ How to Use Your Macintosh
- ☐ How to Use Your Apple IIe

ST

My computer uses:
- ☐ PC DOS
- ☐ MS DOS
- ☐ CP/M

Enclosed is $75 each plus $2.50 shipping (CA add 6.5%)

Name____________________

Phone (_____)____________________

Street____________________

City____________________

State__________ Zip__________

Mail order form to:
Software Training Company
A Division of ATI
12638 Beatrice Street
Los Angeles, CA 90066

For rush order call:
1 (800) 421-4827

VISA MasterCard

ATI Money Back Guarantee
If you're not completely satisfied with how fast you learn with ATI Interactive Training Power Programs, simply return within 3 days for full purchase refund.

ATI Training Programs™

Learn software the natural way.

cluding *device = Ansi.sys* in the Config.sys file. Alas, on reading the fine print, I noticed a reference to a DOS function call to make the escape codes work. I even tried using the BASIC *print* statement, but it had the same effect. I now believe only those knowledgeable in assembly language can take advantage of the escape codes. I'll bet others have experienced this problem and could use an answer if one exists. Thank you.

Scott Chaban

A: The DOS 2.0 manual is rather vague in this area, but you're on the right track. You do need the line *device = Ansi.sys* in your Config.sys file. To get the escape character into your file, you can use Debug or BASIC. DOS or Edlin won't work because they interpret the escape character as a line-cancel command.

To create a file using BASIC, type the following lines:

```
A>basica
open "esc" for output as 1
print #1, chr$(27)
close
```

and you'll have a file (named Esc) that contains a usable escape code. Use a text editor to copy the file Esc into another file and then add the rest of the control characters to turn echo off and on.

If you'd prefer using Debug rather than BASIC to enter the escape code, follow the steps listed above in the answer explaining how to set permanent foreground and background colors.

Because BASIC does its own character output, you're going to have to use Ansi.sys from DOS. So if you're writing your program in BASIC, use BASIC's *shell* command to leave BASIC, execute the escape sequence from DOS, and then return to BASIC. Dan Rollins's article on BASIC 2.0's undocumented commands (*Softalk*, January 1984) explains how to make the *shell* command work correctly.

This month we have letters from two readers requesting information *you* may be able to provide. We're publishing their business addresses so you can respond to them directly.

The first is from Frank Caswell, who would like to have a package to manage long-term care facilities. He would like general ledger, accounts payable, accounts receivable, payroll, census, patient data, preparation of medical assistance or welfare billings, and employee data. Please send information to:

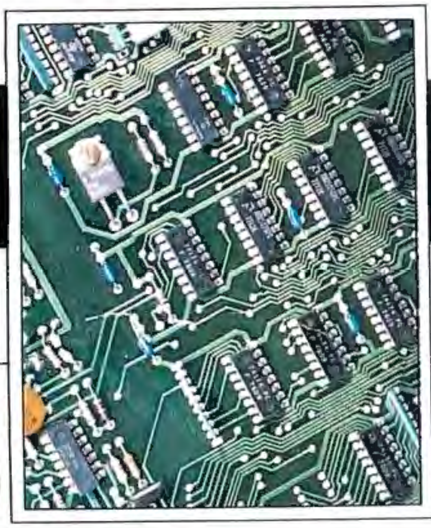
Frank M. Caswell, Jr.
Susquehanna Center for Nursing and Rehabilitation
Box 3245
Harrisburg, PA 17105

The second is from John H. Hess, Jr., who describes *himself* as a "newly minted director of marketing for a small software house." He writes: "We develop, market, and install management information systems for IBM S/34, S/36, and the PC-XT in educational environments. While I am familiar with the general literature on marketing, I find very little relevant to our product or market."

Hess would like suggestions about books or articles on marketing computer software, particularly more expensive customized packages, and databases that could be searched for relevant information on this subject.

Readers with this information may send it to:

John H. Hess, Jr.
Box 386
Harrisonburg, VA 22801 ▲



BOARDS AND BUSES

by Kevin Goldstein

The Maynard Electronics Ten-Megabyte Hard Disk

Part of the pleasure of using a PC with the ten-megabyte hard disk subsystem from Maynard Electronics (Casselberry, FL) is that such a PC can boot directly off the Maynard disk. To those of you not using hard disks, this concern about whether a hard disk system is self-bonting or not must seem a little puzzling. But a hard-disk owner, after shelling out anywhere between one and three grand for a Winchester, doesn't ever want to think about floppies again, let alone depend on them for booting.

A secondary advantage of self-booting hard-disk systems is that they conserve desk space. Since you use the floppy drive so infrequently, it becomes quite practical to excommunicate the entire system unit to some unused corner of the floor under your desk. You can even set the system unit on its ear on the floor, with the power switch facing up.

Because the disk installs *inside* the PC system unit, a Maynard-equipped PC acts almost like an XT (almost, but not quite). It even looks like an XT; the only visible difference is trivial: The Maynard-equipped PC has an extra light on the front panel. There's a nontrivial audible difference, however: The Maynard unit is noticeably quieter than your typical XT.

The reason the Maynard disk coexists so peaceably with a PC is that the company is using half-height drives. While the savings in space don't matter—there's no way to make use of the space that's been opened up above and below the drive—the physically smaller unit gulps less power, thereby overcoming the principal objection to installing a hard disk in a standard PC (an XT's power supply cranks out about twice as much juice as a PC's).

The Maynard does conserve space inside the system unit. Because the controller board that comes with the system is itself designed to accept other special function boards, you can actually free up slots when you add the Maynard hard disk controller.

In principle, Maynard's system is similar to the LNW Busboard (reviewed in the October 1983 issue of this column). In practice, it's actually a little nicer, because the modules install more easily and the board itself is a better-looking, more professional piece of work.

A Maynard hard disk system can be assembled from any of three basic configurations. The cheapest buy would be a bare multifunction card, with only the hard disk controller module installed. The hard disk controller is a "three-wide" affair, giving you the potential of installing up to three other modules at a later time. Parallel and serial ports, clock/calendar, and game adapter modules are currently available, and you can expect to see that list of options grow.

If your system unit is already crammed to the gills, Maynard has two other solutions. The first is a *floppy* disk controller board that accepts three additional modules; since the hard disk controller board is three wide, it eats up all three of those spaces. If you opt for this setup, though, you can throw away your IBM floppy disk controller card, because the floppy controller portion of Maynard's board is an exact replacement.

Well, no, not really an *exact* replacement. The IBM board allows you to install up to four 5 1/4-inch floppy drives, whereas the Maynard board lets you install four 5 1/4-inch floppies or two 5 1/4-inchers and two 8-inchers.

(The 8-inch-drive capability could be particularly useful in the coming months, since the new super-high-capacity 5 1/4-inch drives just appearing on the market are likely to use the older [and higher-performance] 8-inch interface. Drivetec, for example, now has a 3.3-megabyte 5 1/4-inch floppy drive that uses the 8-inch interface. Backing up a 10-megabyte drive to three 3.3-megabyte super-floppies would be a lot less painful than shuffling thirty ordinary floppies.)

If the idea of junking your floppy controller board doesn't turn you on, but you would like to add more memory (and who wouldn't?), Maynard's third option may appeal: a multifunction card with sockets for up to 576K. The hard disk controller module is installed *over* the RAM and eats up all three available module spaces.

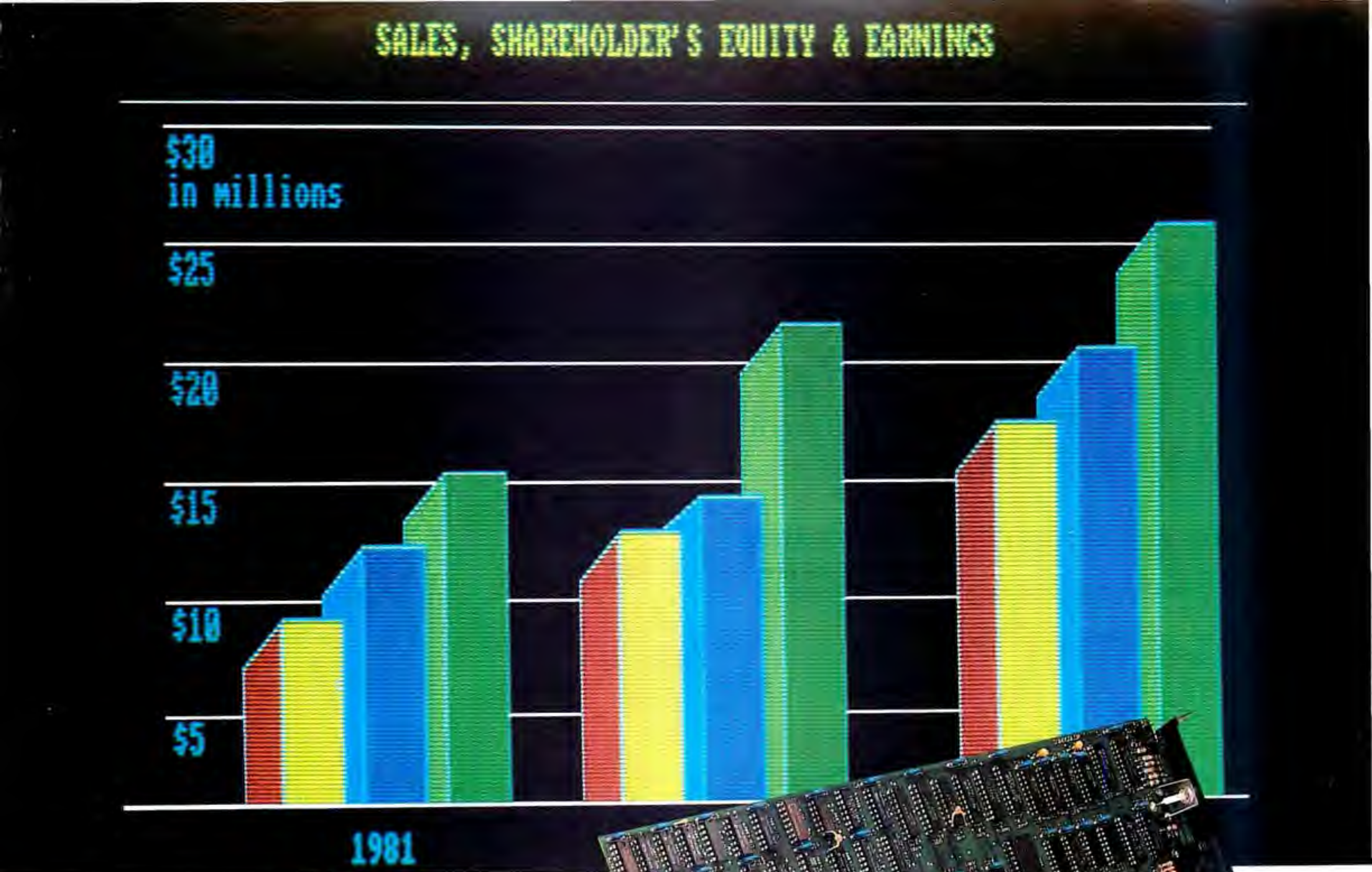
In operation, Maynard's subsystem is a good deal more "XT-like" than, say, the Pegasus system ("Boards and Buses," November 1983), but it still falls short of 100 percent XT-compatibility. Maynard has designed its own controller card, which is not a functional clone of the Xebec board used in the XT. That means that any software that veers around the ROM BIOS and writes directly to the controller-board hardware doesn't work with Maynard's hard disk. You're not likely to encounter such problems unless you try to run foreign (non-PC-DOS) operating systems.

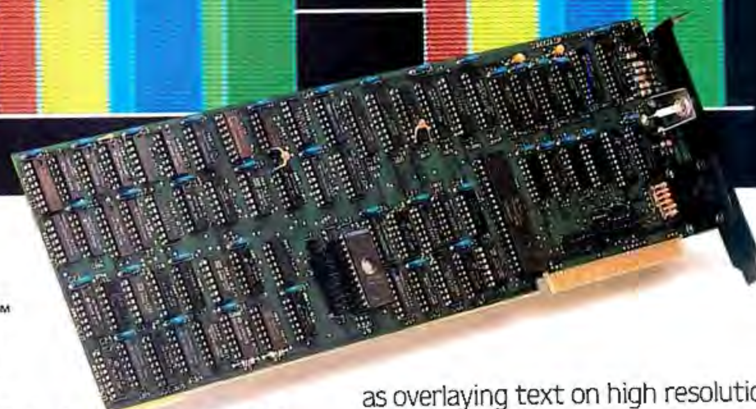
It remains to be seen whether PC/IX, IBM's newly announced version of Unix, goes through the ROM BIOS or fiddles directly with hardware. If PC/IX pulls a fast one and writes direct to the disk controller hardware, it should be fairly simple to write an installable device interfacing PC/IX to the Maynard hard disk (or to any other hard disk, for that matter).

That is just what Kim Knapp, founder and president of Maynard, intends to do. Judging by the firmware the company has already written for its machine, Maynard should have no problem cranking out a PC/IX device driver.

That firmware, which is delivered with each system in the form of a ROM, is the key to the hard disk's self-booting ability. The ROM plugs into the unused ROM socket on the PC motherboard, where it is automatically scanned as a part of the power-up routine. Detecting the presence of a ROM, the boot routine gives that ROM control of the machine; the firmware in the Maynard-supplied ROM causes the disk-reading procedures of the booting process to access the hard disk instead of the A drive. The ROM contains all the routines needed for accessing the hard disk, which makes loading an installable device driver unnecessary.

(The system is not self-booting when used with older PCs that have 64K motherboards, or with older Compaqs, since these lack the empty ROM socket. Much to their credit, Compaq recently added the empty ROM





QUADCOLOR™ BY QUADRAM™

Enhance the graphics capabilities of your IBM Personal Computer and XT. With Quadcolor by Quadram. The color graphics adapter set packed with bright ideas about color and graphics.

BRILLIANT COLOR AND GRAPHICS FOR YOUR IBM® PC OR XT.

Begin with Quadcolor I
Snap Quadcolor I into your PC to drive any type color monitor, even your home television set (with an RF modulator). Then take advantage of Quadcolor I enhancements to brighten up your program work. Enhancements like 32K of on-board memory, for up to 16 active video pages in 40-column text mode. Or 8 pages in 80-column. (That's twice what other popular cards offer.) Switch to graphics mode and create two complete screen pages in memory. Edit them separately for Quadcolor I special effects like animation.

Plus, Quadcolor I gives you two true colors in high resolution mode. So there's no need to limit yourself to just black and white anymore.

Expand with Quadcolor II
As your needs grow, expand to Quadcolor II with an upgrade board and discover the brilliance of *advanced* color graphics. Like a full 16 color high resolution mode and an amazing *136 color* medium resolution mode.

Plus Quadcolor II comes complete with BASICQ, the software package that enhances IBM's BASICA graphic commands to increase your programming options (such as overlaying text on high resolution graphics). And when you're in the mood for a little fun, Quadcolor II provides a game port for your use.

Pick the set that's packed...
Only Quadcolor packs so many bright ideas into one color graphics adapter set. Backed by the solid tradition of Quadram Quality. Visit your local retailer soon for a demonstration, then picture it for yourself on your own IBM PC or XT. You'll agree. For brilliant, advanced color graphics capabilities, it's Quadcolor by Quadram. A bright idea whose time has come.

QUADRAM™ QUALITY

QUADRAM™
An Intelligent Systems Company

4355 International Blvd./Norcross, Ga. 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

socket to their boards, so the Maynard system can be made self-booting with Compaqs fresh off the assembly line.)

The result of all this hoopla is a machine that boots directly off the hard disk, just as an XT does, unless you've got a bootable disk in the A drive; in the latter case, the machine boots from the A-drive floppy—again behaving just like an XT.

In operation, the disk is everything you'd want it to be: quiet, fast, reliable, and unobtrusive. As a matter of fact, the only serious problem is—or, more correct, *was*—the manual. According to Knapp, Maynard is well aware of the shortcomings of its original manual, and by the time you read this, it should be shipping a totally rewritten, typeset manual. It can't be any worse than the first one, and it might be a lot better.

Here's a suggestion that may or may not appear in the new Maynard manual.

The disk is formatted by the IBM-supplied *fdisk* command. Both the IBM and Maynard hard disks are configured as 306 cylinders. (If you were to stack two floppy disks on top of one another, the combination of track 22 on the upper floppy sitting directly over track 22 on the lower could be called *cylinder 22*. On a hard disk, a cylinder simply refers to all the like-numbered tracks on each platter.) The Xebec controller supplied by IBM uses the 306th cylinder to write diagnostic information, whereas with the Maynard controller that cylinder is available for data. If you use *fdisk* on an XT and tell it to use the entire disk for the DOS partition, you end up with a 305-cylinder DOS partition; when you do the same with a Maynard system, you end up with a 306-cylinder DOS partition. Since certain programs—the *Norton Utilities*, for example—don't work on a disk of other than 305 cylinders, you should format your disk to use only 305 for the DOS partition. You can do that by answering "no" to the question:

Do you wish to use the entire fixed disk for DOS [Y/N] .... ?

Then tell *fdisk* to use only 305 cylinders. Any time you create more than one partition on a fixed disk—and that's exactly what you'd be doing here, even though the second partition would be only one cylinder—DOS no longer knows which partition is bootable. For that reason, when you're done formatting, choose item 2—"Change Active Partition"—from the *fdisk* menu to tell DOS that the 305-cylinder partition is bootable.

Because the Maynard disk installs in the system unit, thereby saving the cost of an extra power supply, chassis, and cabling, you might expect it to go for a little less money than an equivalent external system. You'd be right. The whole shebang, with a multifunction card, hard disk controller, drive, and ROM, retails for $1,400. ▲

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • KAYPRO™ • CP/M™ • COMMODORE 64™

The VERSABUSINESS™ Series

Each **VERSABUSINESS** module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS™** module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS™** module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:
Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

COMPUTRONICS INC

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

EXEC DAVONG:

DRY CLEANING'S LOSS

It's a myth. There is no Mr. Davong.

There is, however, a John Davilla and a Tom Hong. Davilla put up the money that launched Davong, one of the leading players in the IBM hard-disk market, but Hong's the prime mover of the company.

In many ways, the company that Thomas Hong founded in 1981 is the archetypical Silicon Valley company. Housed in a high-tech building just off Highway 101, the tie line for hundreds of fast-track electronics companies—firms with famous names like Intel, National, Hewlett-Packard, and Fairchild—Davong went from no income to annual revenues of over $35 million in well under two years.

Which is not to say that Hong hasn't paid his dues; the road that led to success was longer for him than for some of the other legends of Silicon Valley. By Valley standards, Hong is almost an old man. On the other hand, he's hardly ready at age 46 to consign himself to an old-age home.

While Hong's dream of success may have taken a little longer to become reality than he would have liked, those extra years in the electronics industry paid handsome dividends when he did start his own shop. Hong's long career encompasses jobs in a variety of electronics companies, in positions ranging from technician to engineer to manager. That depth of experience gave him two distinct advantages when he formed his own firm.

First, he knew not only what the market wanted but also whether the state of the electronics art was sufficiently advanced to supply it at an acceptable price. Second, he was personally acquainted with most of the major players in the industry. That's an important point, both because his established track record gave him a quick entree to those who could finance his new company and because it enabled him to hire the best and brightest the Valley had to offer.

Hong started on the path to his career in electronics almost accidentally. He was, as he puts it, "the son of a poor Chinese laundry man, and my father's ambition for me was to take over the family laundry business and expand into dry cleaning." It seems that Hong's father's

IS HARDWARE'S GAIN

English was quite limited, a fact that made it impossible for him to pass the tests necessary to get a dry cleaning operator's license. So the plan was that Hong the younger would attend dry cleaning college, where he would learn about all the chemicals and other exotica used in the upscale laundry business.

But young Hong was less than thrilled with the idea of hauling loads of heavy wet laundry and breathing noxious chemicals all day. He decided to go to college instead, much to his father's disappointment.

"Since my father was very poor, I had to find my own way to obtain an education," Hong recalls. He discovered his ticket to education in an apprenticeship program at the Mere Island Shipyards, where high school graduates with no experience were paid a full-time salary while working as apprentices and learning a skill. After carefully examining all the vocations offered, Hong decided on electronics, because, as he tells it, "it was the cleanest and safest, compared to a lot of other jobs where people lost fingers, broke legs, or carried heavy loads."

Although Hong's early interests were aligned more toward physics than electronics—"I'm a real fan of Einstein," he says — an enthusiasm for electronics blossomed during the hands-on experience he got in lab classes.

"I started playing around with electronics at home," he recalls, "building things like hi-fi kits and RF generators. I really became engrossed in the fascination of electronics."

During his two years as an apprentice, Hong managed to save enough money to go to Berkeley, where he completed a four-year electrical engineering program in three years, working simultaneously as an assistant butcher, waiter, janitor, gardener, and at any other job he could get.

After that kind of college experience, Hong's first regular electronics industry job must have seemed like a vacation. He tackled it with vigor. His work on the design of logic circuits for the Apollo program quickly caught IBM's eye, and Big Blue subsequently hired him away.

Hong's new position at IBM proved to be the first of a number of

Tom Hong spent considerable time in both the electronics and computer industries before founding Davong, which now has annual earnings of almost $36 million.

BY KEVIN GOLDSTEIN

Although Tom Hong and John Davilla founded Davong, its success has been a team effort. From left to right: Tom Hong, president, Ron Apt, vice president of hardware engineering, Mike Wohler, director of quality and technical services, Jim Graber, vice president of finance, Chuck DePew, vice president of marketing and sales, Randy Knox, vice president of manufacturing, and Tim Lundeen, vice president of software.

such experiences that would introduce him to the movers and shakers in the electronics industry. For at IBM, Hong was working for Al Shugart—the same man who was later to found first Shugart, the company that launched the small hard-disk market, and then Seagate, the company that launched the 5 1/4-inch hard disk.

In some ways, Silicon Valley's standards are the inverse of the rest of the country's. In most parts of the United States, a professional who changed jobs every year or two would be looked at with suspicion and would find it increasingly hard to get a job. In Silicon Valley, the competition for quality employees is so intense that anyone who doesn't change jobs every year or three is suspect.

In the twenty or so years between his first electronics job on the Apollo program and the launching of Davong in 1981, Hong held engineering and managerial posts at IBM, National Advanced Systems, Intersil, Apple, Memorex, and Commodore, broadening his industry experience and picking up valuable contacts with each move. For example, at IBM he worked with Shugart on mass storage devices; at Memorex he did disk systems; at National, he was in charge of all engineering for add-on memories for IBM 370-series computers; and by the time he got to Intersil, he was a manager with direct responsibility for more than a hundred employees.

In between, Hong had some experiences that were, shall we say, interesting in their own right. There was the time, for example, after he left Intersil and formed his own corporation to do some special consulting work for Mostek (a pioneer in the manufacture of dynamic RAM memories).

"I was trying to get them into the IBM memory add-on business; it would have been a natural for them, because they were very big in 16K RAM production," Hong recalls. It was also a task that was a natural for Hong: In light both of his years at IBM and of his previous experience developing IBM add-on memory at Intersil, he was particularly well positioned to propel Mostek to success in the memory add-on market. It could have been a partnership made in heaven, as Mostek recognized in a contract with Hong that promised him a percentage of the royalties on all sales for future years. Unfortunately, things didn't turn out quite as Hong expected: Mostek was just at the point of being bought by United Technologies and didn't want to bring the uncertainty of a major new business venture into the merger. They scrapped the deal before it was even started.

"So there I was, thinking I was about to be financially secure, and all of a sudden my potential for royalties disappeared, and I found myself ... somewhat retired instead. Unemployed!"

But someone of Hong's energy, intelligence, and experience doesn't remain unemployed for long. It seems that Apple was looking for a division manager. When Hong applied, though, Apple realized they had a more pressing need. They fired the manager of the Lisa project and hired Hong.

Though Hong was happy and doing well at Apple—"I'm one of the

few managers that managed to stick to the schedule," he claims proudly—six months later, as he tells it, "I had the misfortune to be contacted by Commodore and offered a deal just too good to refuse."

Misfortune? Hong failed to hit it off with Commodore's Jack Tramiel and left after three months.

If United Technologies's decision to buy Mostek precipitated the demise of Hong's first stab at millionairedom, his move to Commodore was equally catastrophic, at least in retrospect; in doing so he walked out on Apple stock options that would later be worth several million.

Hong's loss turned out to be the IBM PC market's gain. Finding himself again at loose ends and deciding it was time to venture out on his

Mike Wohler, Jim Graber, and Tim Lundeen are key figures in Davong's ongoing success.

own, he came up with a plan to manufacture a portable personal computer—a revolutionary idea, considering that the Osborne I hadn't yet hit the market.

Unfortunately, not quite revolutionary enough: When he approached venture capitalists with his new idea, he found that Mr. Osborne had already knocked on the same doors and had beaten him to the money. Aced out again, it appeared.

But, ever resilient, Hong took another look at the market, this time focusing on peripherals.

"As I looked at the companies in the market," he explains, "I saw a large number that did a small amount of business." At the time, Hong saw only one company that offered the three related functions he believed would be critical in the coming years: hard-disk systems, hard-disk backup, and local networking.

"I felt there needed to be another company to challenge Corvus along those lines, and I felt that a new company could very successfully challenge Corvus if it had significantly higher-skilled people," he says. It was a philosophy born of his years in the industry and in particular his time at IBM, for it is IBM's theory that a company's success hangs on the success of each individual employee.

Once Hong had his product line firmly in mind, his time in the industry started paying dividends. He began by using his contacts to locate John Davilla, who had a successful PC board operation that was doing business with companies such as Apple, Tandem, Seagate, and Phase Four. Within a week of their first meeting, the two had come to an agreement: Davilla would commit the money, Hong would do the work. (The name took longer to dream up.)

When Hong next turned his attention to the all-important job of staffing, his contacts paid off once again; he is proud of the fact that all the officers of Davong have a B.S. or B.A. and over half have master's degrees. And Hong treats his people well: Stock-option incentives have been known to extend to the secretarial level.

The next order of business was, of course, the product line. From design to manufacture, Davong's first product, an add-on memory board for the PC (guess where he got that idea), was completed in just two months. The company shipped $2,000 worth of orders in January 1982, $8,000 in February, $30,000 in March, and $113,000 in April.

Hong says he chose memory boards as the company's first product because they can be brought to market quickly. Yet his next product, the hard-disk system that cinched the company's reputation, was also completed in only two months. With the recent introduction of Davong's MultiLink local-area network, the third element of Hong's three-point product base is now in place.

It's not easy for companies as successful as Davong to come up with encores that fly as high as the original products. But it looks as though Hong's common sense and engineering talent may have produced a big winner in MultiLink: The product uses the popular and proven Arcnet technology, which has a far larger installed base than Ethernet, notwithstanding all the brouhaha about the latter. Further, the Arcnet technology is significantly cheaper than Ethernet technology, an advantage for which it pays a relatively minor price in performance.

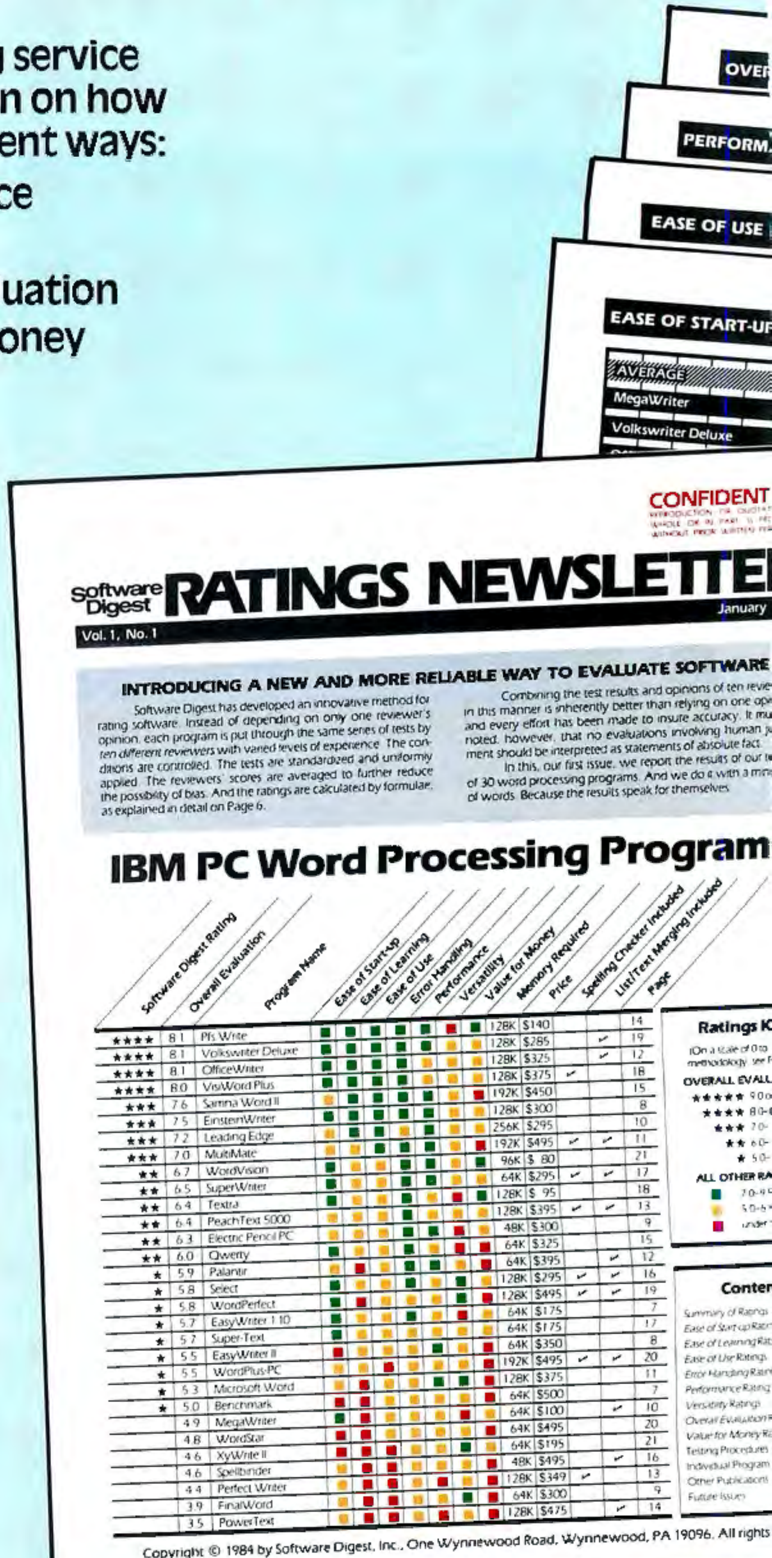
Whether or not MultiLink becomes as big a success as Davong's other products—and there is every reason to think it may—the company is by now on solid footing. By June 1982, barely six months after their doors opened, Davong's revenues hit half a million dollars. They now stand at roughly $36 million a year. The final piece de resistance could come this year, if, as Hong hopes, the company is taken public.

Sometime between the add-on boards and MultiLink, Hong's father forgave him for not going into the dry-cleaning business. ▲

Software Digest introduces THE RATINGS NEWSLETTER™

A new, hard-nosed testing and rating service to give you rapid, reliable information on how IBM PC programs compare in 8 different ways:

- ✓ Ease of Start-up
- ✓ Ease of Learning
- ✓ Ease of Use
- ✓ Error Handling
- ✓ Performance
- ✓ Versatility
- ✓ Overall Evaluation
- ✓ Value for Money



Software Digest RATINGS NEWSLETTE

January

Vol. 1, No. 1

INTRODUCING A NEW AND MORE RELIABLE WAY TO EVALUATE SOFTWARE

Software Digest has developed an innovative method for rating software. Instead of depending on only one reviewer's opinion, each program is put through the same series of tests by ten different reviewers with varied levels of experience. The conditions are controlled. The tests are standardized and uniformly applied. The reviewers' scores are averaged to further reduce the possibility of bias. And the ratings are calculated by formulae, as explained in detail on Page 6.

Combining the test results and opinions of ten revie in this manner is inherently better than relying on one op and every effort has been made to insure accuracy. It mu noted, however, that no evaluations involving human ju ment should be interpreted as statements of absolute fact.

In this, our first issue, we report the results of our te of 30 word processing programs. And we do it with a mini of words. Because the results speak for themselves.

IBM PC Word Processing Program

Software Digest Rating	Overall Evaluation	Program Name	Memory Required	Price	Spelling Checker Included	List/Text Merging Included	Page
★★★★	8.1	Pfs Write	128K	$140			14
★★★★	8.1	Volkswriter Deluxe	128K	$285		✓	19
★★★★	8.1	OfficeWriter	128K	$325		✓	12
★★★★	8.0	VisiWord Plus	128K	$375	✓		18
★★★	7.6	Samna Word II	192K	$450			15
★★★	7.5	EinsteinWriter	128K	$300			8
★★★	7.2	Leading Edge	256K	$295			10
★★★	7.0	MultiMate	192K	$495	✓	✓	11
★★	6.7	WordVision	96K	$ 80			21
★★	6.5	SuperWriter	64K	$295	✓	✓	17
★★	6.4	Textra	128K	$ 95			18
★★	6.4	PeachText 5000	128K	$395	✓	✓	13
★★	6.3	Electric Pencil PC	48K	$300			9
★★	6.0	Qwerty	64K	$325			15
★	5.9	Palantir	64K	$395		✓	12
★	5.8	Select	128K	$295	✓	✓	16
★	5.8	WordPerfect	128K	$495	✓	✓	19
★	5.7	EasyWriter 1.10	64K	$175			7
★	5.7	Super-Text	64K	$175			17
★	5.5	EasyWriter II	64K	$350			8
★	5.5	WordPlus-PC	192K	$495	✓	✓	20
★	5.3	Microsoft Word	128K	$375			11
★	5.0	Benchmark	64K	$500			7
	4.9	MegaWriter	64K	$100		✓	10
	4.8	WordStar	64K	$495			20
	4.6	XyWrite II	64K	$195			21
	4.6	Spellbinder	48K	$495		✓	16
	4.4	Perfect Writer	128K	$349	✓		13
	3.9	FinalWord	64K	$300			9
	3.5	PowerText	128K	$475		✓	14

Ratings K

OVERALL EVALU

ALL OTHER RA

Conten

CONSISTENT TESTS

Competing programs are all tested exactly the same way, with realistic tests designed specifically for each application.

MULTIPLE TESTING

The tests are repeated 10 times for each program, by 10 individual testers with a typical range of backgrounds and varied levels of experience.

UNBIASED RATINGS

The test scores are averaged and the ratings are determined mathematically by computer.

CONCISE REPORTING

The test results and ratings are presented concisely. With a minimum of words. And no advertising. Just useful, bottom-line information from the most methodical, consistent testing organization in the field.

EASY REFERENCE

All reports about competitive programs are in the same issue. Easy to find. Easy to understand.

TEN IMPORTANT ISSUES IN 1984

Each issue focuses sharply on one category of IBM PC software. And these ten categories will be completely retested at least once a year:

1. Word Processing
2. Data Management
3. Spreadsheets
4. Graphics
5. Time Management
6. Self-Improvement
7. Tax & Investment
8. Communications
9. Special Applications
10. Integrated Software

Subscribe today, and you'll know more about the comparative merits of IBM PC programs than even the top experts knew just a few weeks ago.

FIRST ISSUE FREE!

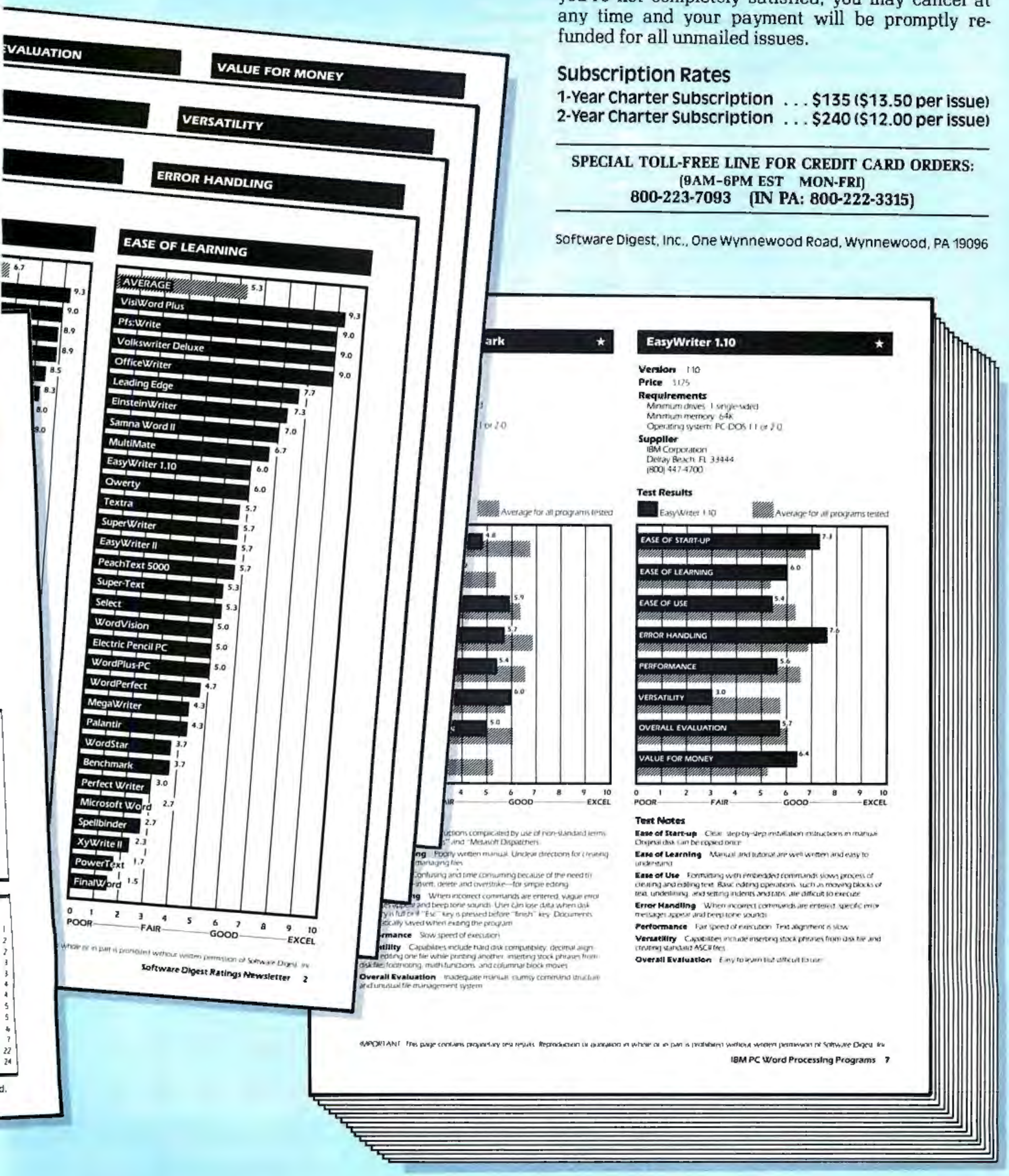
Enter a 1-year or 2-year subscription now, and we'll send you our *complete word processing issue* FREE. Your paid subscription will begin with the second issue, covering data management programs. If you're not completely satisfied, you may cancel at any time and your payment will be promptly refunded for all unmailed issues.

Subscription Rates

1-Year Charter Subscription . . . $135 ($13.50 per issue)
2-Year Charter Subscription . . . $240 ($12.00 per issue)

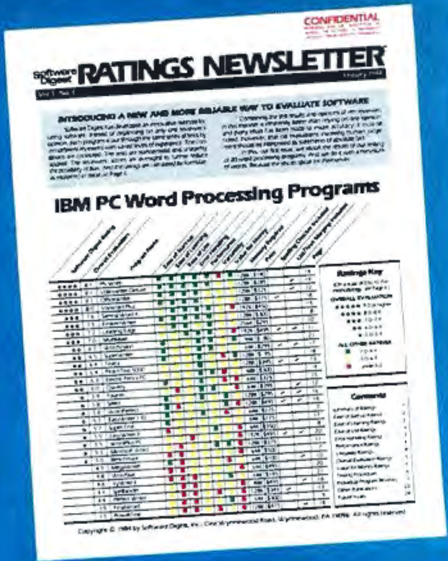
SPECIAL TOLL-FREE LINE FOR CREDIT CARD ORDERS:
(9AM–6PM EST MON-FRI)
800-223-7093 (IN PA: 800-222-3315)

Software Digest, Inc., One Wynnewood Road, Wynnewood, PA 19096



Want to know *how* Software Digest ratings are determined? Please turn the page ►

SUBSCRIBE TODAY AND GET YOUR FIRST ISSUE FREE!



Ten Big Issues in 1984

1. Word Processing
2. Data Management
3. Spreadsheets
4. Graphics
5. Time Management
6. Self-Improvement
7. Tax & Investment
8. Communications
9. Special Applications
10. Integrated Software

✓ The most realistic testing.

✓ The most useful comparisons.

✓ The most reliable ratings.

Software Digest RATINGS NEWSLETTER™

Yes, I would like to receive the results of your extensive tests of IBM PC programs. Enter my subscription to the Software Digest RATINGS NEWSLETTER, and send my FREE COPY of the first issue right away. If for any reason I am not satisfied after receiving the second issue, I may cancel at any time and you will send me a prompt refund for all unmailed issues.

☐ 1 Year (10 Issues) for $135 ☐ 2 Years (20 Issues) for $240

(Overseas orders must add $3 per issue and be fully prepaid in US currency.)

☐ Payment enclosed ☐ Bill me ☐ Bill my company

Charge to my ☐ MasterCard ☐ VISA ☐ AmericanExpress

Card No. Exp. Date

Name__________

Company__________

Address__________

City, State, Zip__________

Signature__________

SPECIAL TOLL-FREE LINE FOR CREDIT CARD ORDERS: 800-223-7093 (IN PA: 800-222-3315)

SFSK044

Software Digest RATINGS NEWSLETTER™

Yes, I would like to receive the results of your extensive tests of IBM PC programs. Enter my subscription to the Software Digest RATINGS NEWSLETTER, and send my FREE COPY of the first issue right away. If for any reason I am not satisfied after receiving the second issue, I may cancel at any time and you will send me a prompt refund for all unmailed issues.

☐ 1 Year (10 Issues) for $135 ☐ 2 Years (20 Issues) for $240

(Overseas orders must add $3 per issue and be fully prepaid in US currency.)

☐ Payment enclosed ☐ Bill me ☐ Bill my company

Charge to my ☐ MasterCard ☐ VISA ☐ AmericanExpress

Card No. Exp. Date

Name__________

Company__________

Address__________

City, State, Zip__________

Signature__________

SPECIAL TOLL-FREE LINE FOR CREDIT CARD ORDERS: 800-223-7093 (IN PA: 800-222-3315)

SFSK044

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 526 WYNNEWOOD, PA

POSTAGE WILL BE PAID BY ADDRESSEE

Software Digest
One Wynnewood Road
Wynnewood, PA 19096

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 526 WYNNEWOOD, PA

POSTAGE WILL BE PAID BY ADDRESSEE

Software Digest
One Wynnewood Road
Wynnewood, PA 19096

THE RATINGS NEWSLETTER™

A new, hard-nosed testing and rating service to give you rapid, reliable information on how IBM PC programs compare in 8 different ways:

- ✓ Ease of Start-up
- ✓ Ease of Learning
- ✓ Ease of Use
- ✓ Error Handling
- ✓ Performance
- ✓ Versatility
- ✓ Overall Evaluation
- ✓ Value for Money

CONSISTENT TESTS

Competing programs are all tested exactly the same way, with realistic tests designed specifically for each application.

MULTIPLE TESTING

The tests are repeated 10 times for each program, by 10 individual testers with a typical range of backgrounds and varied levels of experience.

UNBIASED RATINGS

The test scores are averaged and the ratings are determined mathematically by computer.

CONCISE REPORTING

The test results and ratings are presented concisely. With a minimum of words. And no advertising. Just useful, bottom-line information from the most methodical, consistent testing organization in the field.

EASY REFERENCE

All reports about competitive programs are in the same issue. Easy to find. Easy to understand.

TEN IMPORTANT ISSUES IN 1984

Each issue focuses sharply on one category of IBM PC software. And these ten categories will be completely retested at least once a year:

1. Word Processing
2. Data Management
3. Spreadsheets
4. Graphics
5. Time Management
6. Self-Improvement
7. Tax & Investment
8. Communications
9. Special Applications
10. Integrated Software

Subscribe today, and *you'll know more* about the comparative merits of IBM PC programs than even the top experts knew just a few weeks ago.

Here's how Software Digest tested and rated 30 IBM PC word processing programs—a project that took two months and cost over $100,000. This was followed by the testing of data management programs with an even more comprehensive series of tests designed just for that application. Each category of software will be tested with the same methodical thoroughness, consistency and impartiality.

Reproduced from the first issue of the Software Digest RATINGS NEWSLETTER.

THE SOFTWARE DIGEST RATINGS METHODOLOGY

Testing and Rating Procedures for Word Processing Programs

Testing Procedures

Our testing procedures for word processing programs consisted of three phases:

In the first phase (INSTALLATION), the reviewers installed the program following the instructions given by the software supplier. This generally included making one or more backup copies of the master disk, adding the operating system to the program disk, creating a data disk, and selecting the correct type of monitor (color or monochrome) and printer. The installation procedure was timed and comments noted.

During the second phase (THE "2-HOUR TEST"), reviewers were given exactly two hours to learn how to edit a simple letter with the program, plus whatever else could be learned within that period. This was the only phase with a specific time limit. A two-hour limit was chosen because it is about the maximum time a typical user would spend with a program before trying to use it.

In the final phase (EDITING/PRINTING), reviewers were given a prepared disk file containing a standard one-page letter with text to be changed, deleted, moved, copied, centered, underlined, and then printed. As with the installation procedure, editing and printing were timed. Thus, the test was designed to answer the question: "After spending two hours to learn this word processing program, how difficult and time consuming is it to produce a typical business letter?"

At the end of each phase, reviewers reported their observations on a detailed report form. Individual scores were recorded for the ease of performing a typical range of editing functions, as well as ease of installation, ease of learning, ease of use, speed of execution, error messages, error resistance, and overall evaluation.

Rating Procedures

Information provided by the tests was statistically summarized in eight rating categories described below. Each category had a possible score between 0 and 10 (10 being the highest rating).

Ease of Start-up Ease of getting the program "up and running." It included the average time required for installation (inverted and adjusted for upper and lower limits) and the average Ease of Installation score.

Ease of Start-up = ((10 − ((Installation Time − 5) / 45 ∗ 10)) + Ease of Installation) / 2

Ease of Learning The average Ease of Learning score. It was a measure of how simple it was to learn to use the program, as determined by each reviewer.

Ease of Learning = Total Ease of Learning Scores / 10

Ease of Use Statistical mean of the average time required to edit and print the standard letter (inverted and adjusted for upper and lower limits), the average Ease of Use score, and the average Functional score (composed of a series of editing procedure scores).

Ease of Use = ((10 − ((Minutes to Edit − 15) / 105 ∗ 10)) + Functional Score + Ease of Use) / 3

Error Handling A weighted composite of reviewer scores (50%) and our Technical Department's error-testing procedures (50%). Reviewer scores included whether the program "froze" (i.e., refused to respond to further user commands), whether data was lost, and the average scores for Error Messages and Error Resistance. The Technical Department procedures included tests for how the program handled specific situations: disk full, disk directory full, attempting to read a disk with the drive door open, reading and writing to a non-existent drive (e.g., Q:), writing to a write-protected disk, system memory full, printer cable disconnected, and printer turned off. Each program was also checked to see if it prompted users when they attempted to leave the system before saving the current text to disk.

Error Handling = (Freeze + Data Lost + Error Messages + Error Resistance + (4 ∗ Lab Score)) / 8

Performance Statistical mean of the average Execution Speed score and the average letter score (i.e., how well the reviewers were able to correctly edit and print the final copy of the letter).

Performance = (Execution Speed + Letter Score) / 2

Versatility The program's power, above and beyond simple word processing functions, based on a checklist of 20 advanced features. One point was given for each of the 20 features on the following list which the program possessed, and the total was divided by two to place it on a 10-point scale.

Versatility = (Five or more printers specifically supported +
Hard disk compatibility +
User can reassign keys or build macro commands +
Shows preview of printing +
Decimal alignment +
Can edit one file while printing another +
Can insert stock phrases from disk file +
Deletions can be restored +
Creates standard ASCII files +
Split screens or windows +
Multicolumn formatting +
Virtual memory +
Footnoting capability +
Mouse compatible +
Math capability +
Index and/or table of contents creation +
Columnar block moves +
Soft hyphenation +
Text can be more than 80 columns wide +
Print selected page or range of pages) / 2

Overall Evaluation A weighted average of all of the preceding ratings and the average Overall Evaluation score, as follows:

Overall Evaluation = ((2 ∗ Ease of Use) + Ease of Start-up +
(2 ∗ Ease of Learning) + Performance +
(2 ∗ Error Handling) + Versatility +
(2 ∗ Overall Evaluation)) / 11

Value for Money An indication of what to expect for your money. The two variables were the Overall Evaluation rating and the program price. For the purpose of fair comparison, the retail price of each program was adjusted by subtracting $50 for each major accessory program included (such as list/text merge and spelling checker programs). These accessory programs were not tested and, therefore, played no part in the reviewer scores.

Value for Money = (Overall Evaluation + (10 − (Adjusted Price − 50) / 45)) / 2

For more information about The Ratings Newsletter, please refer to the preceding two pages.

SYSTEM Notebook

by Alan Boyd

DOS 2.0's Improved Batch Commands

As we promised last month, we're now going to look at DOS 2.0's improved batch commands. DOS 2.0 brings with it a whole slew of new batch commands and capabilities, adding intelligence and iterative capabilities where there were none before.

We saw in the October and November 1982 installments of this column that DOS 1.1 allows you to store a series of commands in a batch file and execute them by entering a single command—thus allowing you to customize DOS for particular operators and to simplify complex systems. The autoexec batch facility, in addition, enables you to create turnkey application systems that an inexperienced user can bring up simply by turning the computer on.

Batch commands also automate frequently performed tasks so that they can be invoked quickly and easily. Programmers often rely on batch files when they use compilers or assemblers, since the files drastically reduce the number of complex and repetitious keystrokes required to invoke a procedure.

DOS 1.0 has only two commands designed specifically for use in batch files. The first of these is *rem*, which lets you pepper your batch files with remarks; when you come back to the batch file after six months have gone by, your comments will give you a clue as to what you meant the file to do.

The second command is *pause*, which, when placed in a batch file, halts all action and prints the message

    Strike a key when ready

on the screen. You might use this command if you need to stop the batch processing and prompt your user to change disks.

DOS 1.1's batch facility also features replaceable parameters. Replaceable parameters let you define variables in your batch files so that the same batch file, when modified by these variables, can be used for various processes.

Six New Commands. DOS 2.0 has six new batch commands: *echo*, *for*, *goto*, *if*, *shift*, and *cls*. Technically, *cls* isn't a batch command, since it can be used in other DOS applications, but its greatest usefulness is with batch files. Most of the other commands modify the way DOS executes the commands in a batch file, such as allowing conditional execution of commands or varying the number of replaceable parameters that can be used.

Cls. The *cls*, or *clear screen*, command is the simplest new command. It is largely cosmetic—all it does is cause the screen to go blank. However, because DOS 2.0's batch language is much more complex than that of its earlier incarnation, you need something like *cls* to create a neater environment to work in.

Echo. Normally, DOS prints all commands stored in a batch file on the screen as it reads them from a batch file. While having the screen scroll in the manner of a printer may have been just fine for older operating systems, some of DOS's newer capabilities allow very accurate cursor manipulation; therefore, capabilities such as the *echo* command have been added.

Like *cls*, *echo* is cosmetic. Its purpose is simply to enable or inhibit the display of batch commands during the running of a batch file.

The *echo* command has two main modes. These are

    A>ECHO ON

and

    A>ECHO OFF

The former puts DOS in its default state, in which all batch commands are echoed to the screen as they're executed; the latter inhibits this. When you turn the echo off, the commands contained in a batch file aren't displayed, but any command you type at the keyboard is displayed as usual. Turning the echo off lets you set up batch files that have a much more professional look.

If you plan to use this feature, you should consider placing *echo off* as the first command in your batch file. Since it's the first command, it'll be displayed as usual; follow immediately with a *cls* command if you don't want the words "echo off" to be visible on the screen.

Echo affects only the statements in the batch command itself; it has no effect whatever on the screen output from your batch commands. For example, if your batch file includes an *echo off* command followed by a *dir*, you'll see the directory listing on the screen; you just won't see the word "dir."

If you use the *echo* command without an argument, the current status—either *echo on* or *echo off*, is displayed.

One frustration you may have with this command is that *echo* is always turned back on when the batch file finishes; you'll need to put an *echo off* statement at the beginning of each batch file if you want to use it. You can't turn it off once and leave it off. You can see this for yourself by creating a simple batch file containing nothing but the statement *echo off*. Execute the file and then enter *echo*. You'll see that the echo is back on again.

A second mode for using the *echo* command is in sending messages to the screen while the echo is off. When the echo is off, messages contained in *rem* statements and *pause* statements are not displayed. If you want the echo off but want to have a message displayed anyway, you can put that message on the same command line with the word *echo*.

You can get an idea of the way the *echo* command operates by entering and running the following batch file.

```
A>COPY CON: MYTEST.BAT
REM The echo is on,
PAUSE So this message will appear on-screen.
ECHO OFF
REM Now the echo has been turned off,
```

Come in late on "System Notebook"? All back issues of this column—from June 1982—are still available; for further information, see page 4. The columns will also be published soon, as a single volume, by Softalk Books.

PAUSE so this message will not be printed.
ECHO But this one will!
∧Z (F6)

When you invoke this batch file, you should get the following:

A⟩MYTEST
A⟩REM The echo is on,
A⟩PAUSE So this message will appear on-screen.
Strike a key when ready . . .
A⟩ECHO OFF
Strike a key when ready . . .
ECHO But this one will!

Notice the effect of *echo off* on your *pause* line. With the echo turned off, messages associated with *pause* statements are suppressed, but the "Strike a key when ready" appears anyway.

Shift. The *shift* command extends DOS 1.1's replaceable parameter feature, which allows you to assign meanings to ten dummy parameters—%0, %1, %2 . . . %9—from the command line. These dummy parameters are placed in a batch file, and when you invoke the file you replace them with the real parameters. DOS 2.0's *shift* command makes it possible for you to use more than ten replaceable parameters.

Starting with the dummy %1, any string can be assigned to a replaceable parameter; DOS reserves the %0 parameter for the batch filename. People who use the IBM *Macro Assembler* will find this example helpful:

REM This is ASMFILE.BAT
REM First make a backup copy
COPY %1.ASM %2.ASM
REM Next assemble the source file
MASM %2,%2,%2
REM Finally, read the assembler listing
TYPE %2.LST

Instead of entering a long-winded string of commands to make a backup copy of the source file, assemble it, and list it, the entire process can be invoked with the one simple command:

A⟩ASMFILE A:SOURCE B:BACKUP

Ordinarily, the batch filename would be assigned to the %0 dummy parameter, but it isn't used in this example. The parameter A:Source is assigned to %1, and B:Backup is assigned to %2. When these parameters are placed in the batch file, the following results:

REM This is ASMFILE.BAT
REM First make a backup copy
COPY A:SOURCE.ASM B:BACKUP.ASM
REM Next assemble the source file
MASM B:BACKUP,B:BACKUP,B:BACKUP
REM Finally, read the assembler listing
TYPE B:BACKUP.LST

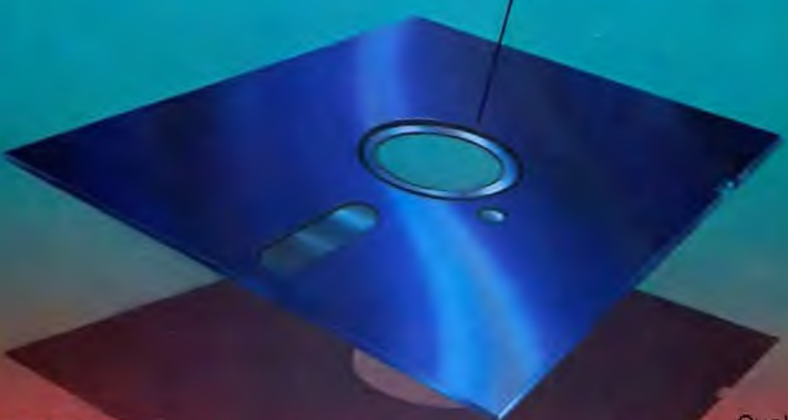
With up to ten replaceable parameters available, complex batch processes can be constructed. For many people, however, ten parameters were just not enough. The solution? The *shift* command, which, when placed in a batch file, changes the order in which the dummy parameters %0 through %9 are replaced.

With each *shift* command, every parameter is shifted to the left. It may sound confusing, but you can easily demonstrate the feature for yourself by preparing a simple example (you'll want to prepare this example with a word processor in ASCII mode or a text editor, since it involves some repetitious typing). The file should look like this:

ECHO OFF
REM This is FUN.BAT
ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9
SHIFT
ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9
SHIFT
ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9
SHIFT

ARE YOU TIRED OF FLOPPY EXCUSES?

"EXCUSE ME, I REALLY HATE TO BOTHER YOU AT A TIME LIKE THIS BUT THIS DISK IS COMPLETELY FULL."



QuadDisk has all the answers
Quadram presents QuadDisk, the Winchester hard disk system that lets you leave all those floppy disks—and their excuses—behind. QuadDisk does everything that floppies do, only better. And faster.

A lot of storage in a little space
For starters, QuadDisk will take care of your storage problems—in more ways than one. QuadDisk comes in 5, 10, 16, 22 and 65 megabyte capacities for all the on-line storage space you need. That's anywhere from about 20 to 225 floppies worth of information. Or up to 36,000 pages of material—and then some.

Removable Cartridge

What's even more amazing is that this powerful hard disk drive simply slips into your second 5¼" floppy drive space to become an integral part of your system.

Ready when you are
It takes just minutes to install and QuadDisk is there to stay. QuadDisk is always there when you need it.

And it works fast. QuadDisk accesses information faster than a floppy. And can perform up to 9 tasks at once.

It comes complete with its own software. And a file utility that guides you when you want to run and manipulate files and programs. There's nothing hard about using this hard disk system. Just follow the simple commands on the screen.

Safety first
QuadDisk protects your work like no floppy can. And for added data security, there's QuadDisk with removable cartridges for up to 5MB on-line mass storage and virtually unlimited off-line storage capacity. Just pop one in when you're ready. Pop it out when you're done.

Quadram Quality
You'll find that QuadDisk is backed by reliable Quadram Quality. It's the mark of dependability and performance from the leader in microcomputer enhancements.

Visit your local computer retailer today and ask to see QuadDisk in action. When you compare it to the floppies, there's no excuse not to buy QuadDisk.

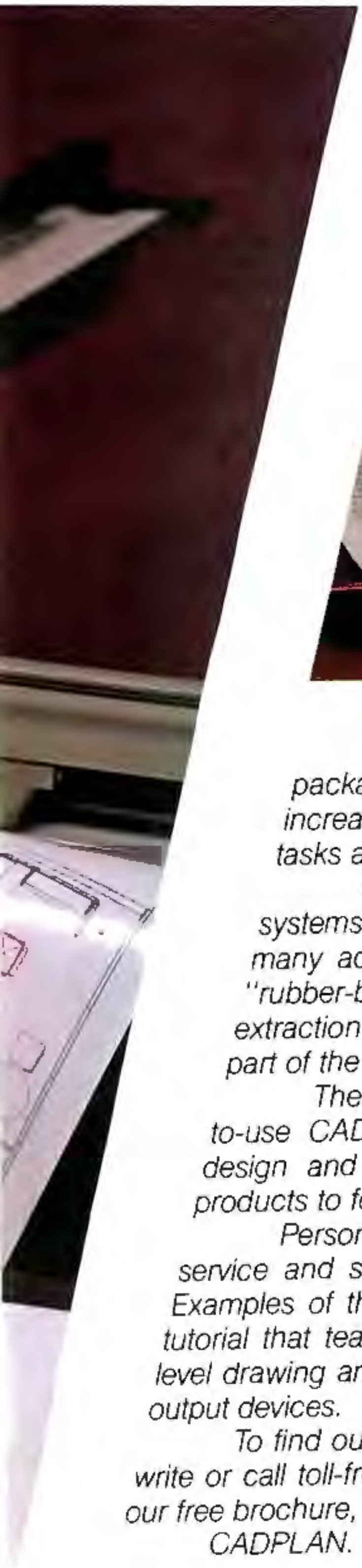
QUADRAM QUALITY™

QuadDisk™ by Quadram.™

QUADRAM™
An Intelligent Systems Company

4355 International Blvd., Norcross, GA 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

CADPLAN
Users Manual
CADPLAN
CADDRAFT
office services
waiting



NOW COMPUTER-AIDED DESIGN COSTS LESS THAN A DRAFTING TABLE.

For only $1,200—less than the price of a top-quality drafting table—you can add computer-aided design (CAD) capabilities to your IBM PC or compatible personal computer.

Introducing CADPLAN,™ a complete user-friendly computer-aided design package for architects, facilities planners, engineers and designers that can increase your productivity by reducing the time you spend on tedious, repetitive tasks and costly design changes.

CADPLAN combines the power and performance of large-scale CAD systems with the ease of use and low cost of the IBM PC. CADPLAN includes many advanced capabilities found in costly CAD systems, such as real-time "rubber-banding," multi-layering and symbol libraries. An optional database extraction feature allows you to perform job costing and analysis as an integral part of the design process.

The CADPLAN design package is part of a growing family of easy-to-use CAD software from Personal CAD Systems, Inc. Software for logic design and printed-circuit board layout is also available, with more CAD products to follow.

Personal CAD Systems is committed to providing the level of after-sales service and support you need to get your job done quickly and efficiently. Examples of this commitment include CADMATE,™ a self-paced, step-by-step tutorial that teaches you how to use CADPLAN, and CADDRAFT,™ an entry level drawing and drafting system. We also support a wide variety of input and output devices.

To find out more about the benefits of low-cost computer-aided design, write or call toll-free 800-882-7535 (in California, 800-635-8306), and ask for our free brochure, "Tools For Modern Design."

CADPLAN. It will revolutionize the way you work and think.

TOOLS FOR MODERN DESIGN™

981 University Avenue
Los Gatos, Calif. 95030
(408) 354-7193 TELEX 278866

p-cad™
PERSONAL CAD SYSTEMS INC.


LIFETREE SOFTWARE
creators of
volkswriter®
THE MOST POPULAR WORD PROCESSOR FOR THE IBM PERSONAL COMPUTER
INTRODUCES
Volkswriter® deluxe
with
HORIZONTAL SCROLLING
TRUE PROPORTIONAL SPACING
STARRING
and a returning cast of
EXTRAORDINARY SIMPLICITY
LIVE TUTORIALS
TEXTMERGE™
plus
over 1,000,000 characters
DYNAMIC
SPECIAL EFFECTS
• COLOR •
• MULTIPLE FONTS •
SPECIAL EFFECTS
• EMPHASIS •
• SUPER/SUBSCRIPTS •
PAGINATION & PROOFING
NOW PLAYING ON THE
IBM/PC and IBM/PC XT
with dBASEII and 1-2-3
COMING SOON:
TI/PC, TANDY TRS-80-2000
only
$285.00
($295 AFTER 2/15/84)
NOW AT YOUR LOCAL STORE OR CALL
(408) 373-4718
™ LIFETREE SOFTWARE

ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9
SHIFT
ECHO %0 %1 %2 %3 %4 %5 %6 %7 %8 %9

With this file we actually enter more than the allowed nine parameters on the command line—we've used shift four times, so we can actually enter thirteen parameters. Okay, now enter the following command—

A>FUN 001 002 003 004 005 006 007 008 009 010 011 012 013 014

—which should result in

A>ECHO OFF
FUN 001 002 003 004 005 006 007 008 009
001 002 003 004 005 006 007 008 009 010
002 003 004 005 006 007 008 009 010 011
003 004 005 006 007 008 009 010 011 012
004 005 006 007 008 009 010 011 012 013

This demonstrates that each replaceable parameter is shifted to the left one place in the priority scheme whenever DOS encounters a *shift* command.

The *shift* command is one of the few completely decadent DOS commands. Few people will ever build batch files that require more than ten parameters to be replaced. However, for those for whom plenty is never enough, there's always the *shift* command.

If. The *if* batch command is probably the most useful new batch feature, because it allows programmers to insert conditional branching into a batch file. Now programmers can build a batch file that makes internal decisions for itself if certain conditions—predefined in the *if* statement—are met.

The *if* command works in much the same manner as the *if-then* statement in BASIC. However, the general Boolean arithmetic conditions that govern the BASIC *if-then* construct don't apply to DOS. Instead, DOS has its own set of conditions:

IF (the condition is true) then do the command X

There are three types of conditions the *if* command can check for. The first of these is *exist*, which checks to see if a specified file is present or not. If the named file is present, then the command that follows it will be executed. If no such file is present, the command won't be executed.

The syntax for such a command would be

IF EXIST MYFILE.TXT COPY MYFILE.TXT MYFILE.BAK

The condition tested by this command is: "Does Myfile.txt exist?" If it does, the statement is true and the condition has been met; the command, in this case *copy myfile.txt myfile.bak*, will be executed. So the *if* command in this case says to DOS, "If the file named Myfile.txt exists, then back it up and call the backup Myfile.bak."

Another form of the *if* command does precisely the opposite. If *not* is placed within an *if* command, then the DOS command that follows will be executed only if the condition is not met. Here's an example:

IF NOT EXIST MYFILE.BAK COPY MYFILE.TXT MYFILE.BAK

This is a handy way for a batch program to determine whether a particular filename is already allocated before it copies another file and uses the name.

With the *if* command, you can also ask DOS to check whether two strings match each other. This way, your batch program can perform functions appropriate to what the operator enters on the command line. The general form of this conditional test is

IF string1 == string2 command

Note the two consecutive equal signs.

One of the things you could do with such a string-check statement is build a minimum-security password scheme into a batch file. The following batch file, for example, includes a secret code that invokes a program

ECHO OFF
CLS
IF %1 == swordfish BASIC PAYROLL
^Z

Your user could then run the payroll program by typing

PASSWORD swordfish

Of course, this example is pretty trivial, and no one with a need for security would ever put together as flimsy a security system as this.

Another use of this version of the *if* command comes in handy if children or other people who are easily intimidated by computers will be using your PC. The *if* command lets you construct a dispatching system that takes a simple command entered by a user and translates it into the complex commands often required to start DOS programs.

If you have a number of programs that are easy to use but difficult for an inexperienced user to start, for example, you might want to place all the programs on a special disk and then construct a batch-file dispatching system to control their operation. Suppose, for example, there are four programs your child likes to play with. Assume that they are a word processor called *Wndrword*, an electronic spreadsheet program called *Wndrcalc*, a drawing program called *Wndrdraw*, and a great game called *Wndrgame*. By putting the following file on the disk with these four programs, you could make it simple for your child to load the right program.

COPY CON: I.BAT
ECHO OFF
CLS
IF %3 == type WNDRWORD
IF %3 == count WNDRCALC
IF %3 == draw WNDRDRAW
IF %3 == play WNDRGAME
^Z

Since this batch file would be looking for a match on the third replaceable parameter, your child could bring up *Wndrgame* by simply typing

I want to play

Doing this would cause the batch file I.bat to be invoked, and the words "want," "to," and "play" to be assigned to the replaceable parameters %1, %2, and %3. DOS would then replace all occurrences of %3 with the word "play" and find the statement

IF PLAY == PLAY

to be true. It would therefore issue the command

WNDRGAME

and run the appropriate program. Similarly, if your child were to enter any other keywords in the right position, DOS would load and run the correct program.

This sample, like the earlier little security system, serves merely as an illustration. It does point out, though, how easy it is to customize batch files using the *if string == string* construct.

The third condition that the *if* command can check for is called *errorlevel*. Some DOS commands, and, in time, many application programs, will support the DOS *errorlevel* scheme. This scheme allows programs that fail to operate correctly to return a code that can be used to determine why the program failed.

On page D-14 of the DOS 2.0 manual, there's a table of error codes available for application programmers. When a correctly written program has finished running, it returns to DOS with an error flag that isn't set. If the error flag isn't set, DOS assumes that all's gone well—that the program has run correctly. If the error flag is set, DOS looks at a specified location to find out what the error code means.

Currently, the only DOS commands set up so that return error codes can be checked are *backup* and *restore*. However, it's a simple matter for programmers to incorporate the error-code system into their programs and thereby make them work with *if errorlevel*.

The syntax for *if errorlevel* is

IF ERRORLEVEL X command

where X is the designated error code. If the returned error code is X or higher, the command is executed. If there's no error code, or if the error

code is lower than X, the command isn't executed.

Since *backup* and *restore* are the only commands that support the error-code system, we'll use them in our example. This sample batch file illustrates the error code's use:

```
BACKUP
IF ERRORLEVEL 1 ECHO - Backup aborted -
```

The *backup* command returns one of four error codes when it exits. If a 0 is returned, no error has occurred. If a 1 is returned, then no qualifying files were found, so no files were backed up. A 3 returned indicates that the user terminated the command with control-break, and a 4 means that another type of error has been encountered.

The above command checks for an error code of 1 or greater. If an error is detected, the "backup aborted" message is printed.

For. The *for* command allows specified DOS commands to be iterated or performed more than once. The full syntax of the command is

```
FOR %%x IN (set) DO command
```

The *for* command is one of the most complex DOS batch commands; it's also one of the most powerful. It allows you to perform the same command on a list of files, directories, or whatever.

The *%%x* is simply a dummy variable associated with each local *for* command. It must consist of a pair of percent signs followed by any valid character other than the digits 0 through 9. If you use one of these numbers, DOS won't be able to distinguish the command from a replaceable parameter. Also, the double percent signs are needed so that after the batch parameter replacement is done there's still one % left. If only one % is included, DOS will look at it when it's in the process of replacing parameters and decide that you'd made a "bad parameter reference" error. It would then throw out the %x and invalidate the *for* command.

The *set* in the command syntax is nothing more than a list of parameters passed to the command, usually consisting of a series of file specifications. The command at the end of the *for* statement is the DOS command to be repeated. As it's repeated, it's applied to each parameter in the set. For example, if you want to type three files on the printer and there's no way to include all three names in a wildcard specification, the usual procedure is to enter three separate commands, something like this:

```
A>TYPE THISFILE.TXT >PRN
A>TYPE THATFILE.BAK >PRN
A>TYPE OTHERFIL >PRN
```

If you look closely at these commands you'll see that we've entered the characters *type* and *>prn* three times. That seems like unnecessary work, but there's no way to shorten the process with wildcards. To make matters worse, we have to sit and wait for each file to be typed before we can enter the next command. Of course, you could always use the PC's typeahead feature, but since you wouldn't be able to see what you were typing, you might make a mistake.

The solution to this problem is to use the iterative capabilities of the *for* command. The following batch command does the trick:

```
FOR %%A IN (THISFILE.TXT THATFILE.BAK OTHERFIL)
DO TYPE %%A >PRN
```

As you can see, the variable we use is *%%A*, and the command is *type >prn*. The set is simply a list of our filenames—Thisfile.txt, Thatfile.bak, and Otherfil.

When you enter the command part of the *for* construct, be careful to include the second *%%x* in the appropriate position. Wherever you want to substitute the name of a file from the *set* in the command, you must use the *%%x*.

One feature that makes this command particularly interesting is that, like all other batch commands, it's really an internal DOS command and can be used on the command line rather than from within a batch file. To use it as an interactive command you simply use a single percent sign instead of a pair.

Goto. The *goto* command is pretty much the icing on the cake as far

as batch programming goes. *Goto* can more than double the capability of the batch language, since it can be used with any combination of commands or conditionals to add flow logic to a batch command.

Goto requires you to use labels in your batch files so that a destination point can be defined. A label is a string of up to eight characters preceded by a colon. In the *goto* command, as in other commands, DOS doesn't differentiate between lowercase and uppercase letters, so that the label *:HERE* is the same as the label *:here*. The following short batch file shows a trivial use of a label.

```
:START
DIR
GOTO START
```

The batch program will ignore the first statement, since it starts with a colon and is therefore a label. It'll proceed to the next statement and issue the *dir* command. When it's finished with that, it'll step to the *goto start* statement. At this point control will jump back to the line that contains the label *:start*—and the procedure will be repeated over and over until you deliberately break out of it by pressing control-break.

When used in conjunction with the *if* command, *goto* provides a whole new level of functionality for DOS batch processing. However, avoid using two labels that have the same first eight characters, since that's all that DOS looks at. If DOS encounters two labels that have the same first eight characters, it will always give precedence to the one you entered first. Therefore, it's best to keep the labels unique and short.

Goto can be made to send control in the batch file either forward or backward. Here's a file that offers an interesting example of the type of functionality that *goto* can add to a batch processor.

```
ECHO OFF
REM This is the file CATALOG.BAT
:START
IF %1 == #1 GOTO DRIVEA
IF %1 == #2 GOTO DRIVEB
IF %1 == . GOTO END
:BUMP
SHIFT
GOTO START
:DRIVEA
DIR A:
GOTO BUMP
:DRIVEB
DIR B:
GOTO BUMP
:END
```

First, DOS stores the file under the name Catalog.bat. You are then given a simple rule. To get a directory listing of any disk drive, you need only start the command with the word *Catalog* and finish the command with the period preceded by a space.

After you've got this file stored on disk, try entering the following commands

```
A>CATALOG THE DISK IN DRIVE #1 .
A>CATALOG THE DISK IN DRIVE #2 .
A>CATALOG THE DISKS IN DRIVES #1 AND #2 .
```

When these commands are entered, the batch file Catalog.bat is executed; then the replaceable parameters feature is used. %0, of course, is the string "CATALOG." The only dummy parameter that we've used in this file, however, is %1, which is assigned the string "THE". The batch file then compares the string "THE" with the strings defined—"#1", "#2", and "."—to see if there's a match. Since there is no match, the batch file continues until it reaches the *shift* command.

When everything is shifted, %1 becomes equal to "DISK" in the first two examples and "DISKS" in the third. The batch file moves on to the command *goto start*. Control is then passed back to the *:start* label, and the comparison tests are run again. Again DOS finds no match, so the *shift* command is activated a second time, causing the %1 parameter to be replaced by the word "IN" in all three cases.

Again, control is returned to the *:start* label and the comparisons are run a third and a fourth time, with no match being found. When *shift* is encountered for the fourth time, however, the parameter %1 is replaced by either "#1" in the first and third examples or "#2" in the second example. Since in these cases the *if* conditions are true, the *gotos* associated with them are executed and control is passed to either the *:driveA* or the *:driveB* label, depending on what the match is.

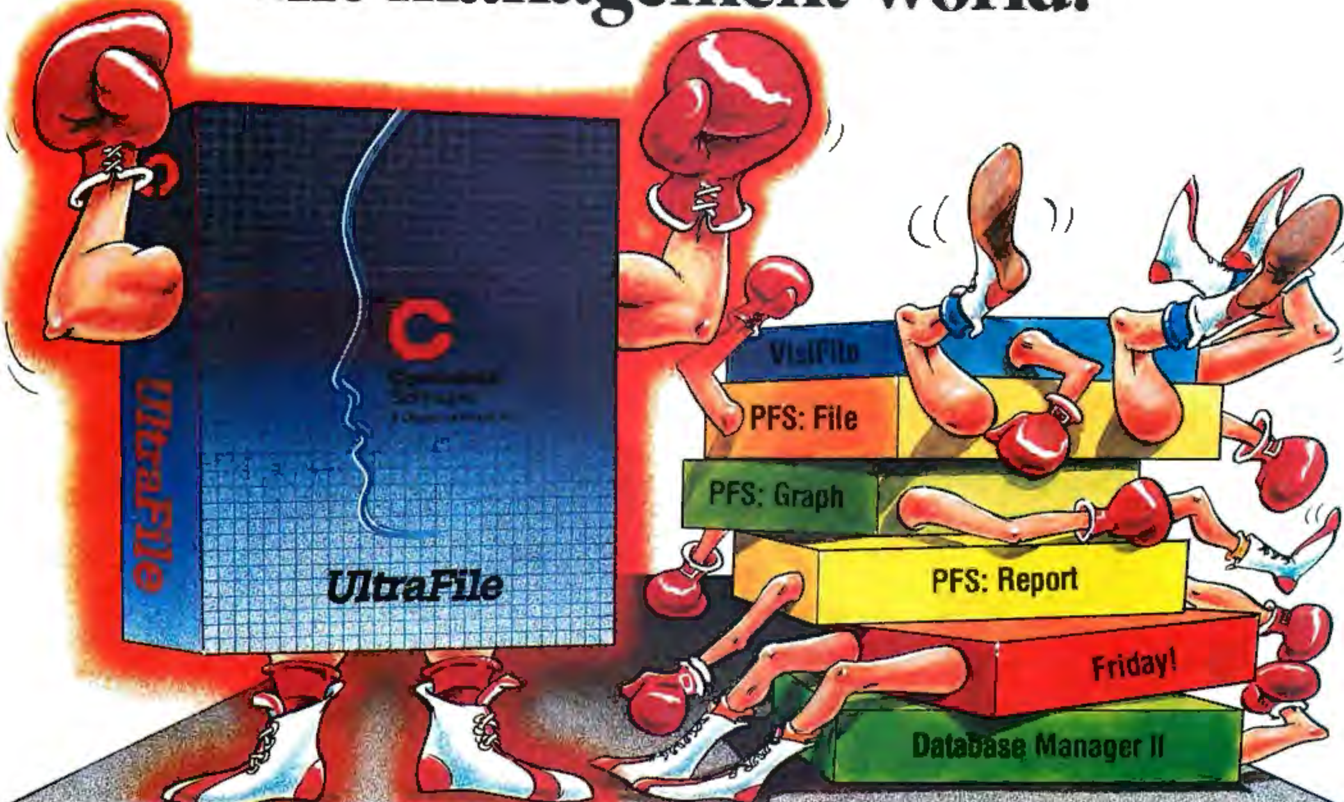
In either case, the appropriate *dir* command is issued, and when that's done, a *goto bump* command is encountered. This sends control back to the *:bump* label, which sends control back to the *shift* command in the initial loop. The process continues in this way, and in the third example another match is found and another *dir* command is issued.

Finally, in each of the three examples the parameters are shifted so that %1 becomes equal to ".", and control is sent via the *if %1 == . goto end* statement to the last statement in the file, which terminates the batch command.

Complex batch commands can be constructed by intelligent use of the *goto* command and labels. When used with conditionals, DOS becomes as powerful as many programming languages, particularly in the control of program flow. Everyone has a different use for these intelligent batch files: Programmers often use them to assist in the construction of program files. Office managers or MIS groups use them to simplify the operation of a complex process for an inexperienced user or even to check users' actions and correct mistakes. Parents and teachers find them useful in simplifying the PC's operation for children.

As you can see, the new batch capabilities of DOS 2.0 go well beyond those of DOS 1.1. These new commands take a little getting used to, but you'll find that they're not as difficult as they may seem. ▲

UltraFile™ -vs- The rest of the database/ file management world.



	PRICE	GRAPHING CAPABILITIES	CUSTOM REPORTS	HELP SCREEN	BROWSE FEATURE	AUTOMATIC DATA FORMATTING	BUILT-IN WHAT IF FUNCTION	ON-SCREEN CALCULATIONS	DATA VERIFICATION	USER DEFINED SCREENS	MULTI-KEY WORD SEARCH
ULTRAFILE	**$195.**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**
Friday!	$295.		X	X		X		X	X	X	
PFS: File, Graph, Rep.	$447.	X	X							X	X
Database Manager II	$295.		X		X			X	X		X
VisiFile	$300.		X							X	X

When you compare features and price, there's no comparison.

If the recent database/file management explosion has left you in a state of confusion, you're not alone. We'd like to clear up some of that confusion.

We've taken the five most popular programs on the market and run a comparison so you can see for yourself exactly where they stand. And what you're getting for your money.

As you can see from the chart, dollar for dollar, feature for feature, UltraFile comes out on top. Here's why:

To begin with, UltraFile has filing, reporting and graphing capabilities all in one package. So, when you buy UltraFile, you automatically get everything you need for creating terrific graphs, custom reports, and managing and storing your information. And you get it all for $195.

Secondly, UltraFile has a help screen. UltraFile leads you step by step through the program. If you need additional help or want to clarify a point, you'll be able to call up the help screen with the touch of a single key. This feature makes UltraFile incredibly easy to use.

Thirdly, UltraFile has automatic data formatting. So if you want a certain word in your text to appear in all caps, you only have to enter it that way the first time. UltraFile automatically capitalizes the word each time it appears. That means fewer errors and more consistency in your text.

UltraFile also does on-screen calculations, has a handy "browse" feature for quickly scanning your data, and has a built-in "what if?" function so you can make projections. Plus, UltraFile talks to the most popular word-processing and spreadsheet programs (1-2-3, WordStar, VisiCalc and others), which gives you greater versatility.

When you stack it up against the rest, UltraFile has the most features for the best price. It just wouldn't make sense to consider anything else.

Available for: IBM PC/XT
Price: $195 complete.
File, Report and Graph.

For the name and location of the UltraFile dealer nearest you, or your free 64 page booklet, "Tips For Buying Software," please write or call: Continental Software, 11223 S. Hindry Avenue, Dept. STI, Los Angeles, CA 90045, 213/410-3977.

Continental Software
A Division of Arrays, Inc.

UltraFile is a registered trademark of Continental Software. Friday! is a reg. t.m. of Ashton-Tate. PFS File, Graph, Report are reg. t.m. of Software Publishing Co. Database Manager II is a reg. t.m. of Alpha Software Corp. VisiFile and VisiCalc are reg. t.m. of VisiCorp. 1-2-3 is a reg. t.m. of Lotus Development Co. WordStar is a reg. t.m. of MicroPro Int'l. IBM PC/XT are reg. t.m. of IBM Corp.

Don't buy a spreadsheet

A lot of electronic spreadsheets just can't cover your needs. They don't go far enough.

They're unable to work like you or adapt to the way you think.

Time for Microsoft® Multiplan,® the high energy spread. A spreadsheet with more workspace than the other leading spread. Full of high performance capabilities for your IBM® PC, PC XT, PC jr. or other MS™-DOS microcomputer.

Multiplan loves you as you are.

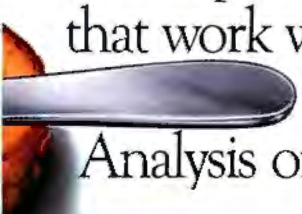
Other spreadsheets force you to learn how they think. Multiplan learns how you think. It remembers the way you work. Anticipates frequent commands. Even offers suggestions on spreadsheet set-up.

Commands are in English. So are formulas. Instead of typing mysterious coordinates like H54 – L73 = BK154, you can simply name worksheet areas: Sales – Costs = Profit.

that spreads too thin.

Multiplan can link information in different spreadsheets. When you make a change on one, every related one is changed.

Multiplan has optional Multiplan Application Programs that work with you to design and build custom spreadsheets for Budget Analysis, Financial Statement Analysis or Cash Planning in minutes. Not hours. Or days.



The trained mouse.

Microsoft's Mouse is a speedy little critter that flies through Multiplan so easily you barely have to lift a finger to select and execute commands. Small wonder Microsoft gets more performance out of a spreadsheet. We designed the MS-DOS operating system that tells the IBM PC how to think. And our BASIC is the language spoken by nine out of ten microcomputers worldwide.

MICROSOFT®
The High Performance Software

To get the best spread call 800-426-9400 (in Washington state call 206-828-8088) for the name of your nearest Microsoft dealer. Now featuring Multiplan at a reduced price: **$195.**

Suggested list price shown for Multiplan.

SOFTALK CLASSIFIED ADVERTISING

Adventure

WIN WITHOUT CHEATING! Top Quality Adventure, Fantasy, Strategy game hints and maps, plus documentation—instruction—command summary. Huge selection. No decoding. Verb-noun list. Refreshing style yet won't give the answer away. $2.50/game. Free catalog. Adventurer's Club, 1673 Via Del Rey, So. Pasadena, CA 91030.

Business

PC PROPERTY MGMT S/WARE

Excellent for owners of rental homes or buildings. Easy to use with clear summaries of cash flows and underlying details. $195 ($95 to dBASE II owners). CA res +6.5%. Call (408) 374-1452 or write Sullivan Assoc., 197 E. Hamilton Ave., #104, Campbell, CA 95008.

STOCK OPTION ANALYZER NEW EDITION

Options-80A, with Black Scholes modeling. Emphasizes return on invstmnt. $170 + $5 P&H. Free brochure. M/C & Visa. Options-80, Box 471-R, Concord, MA 01742.

TAX 4-5-6 (FOR LOTUS 123)

Forms: 1040,A,B,C,D,E,G,R,RP,SE,W,2106,2441, 3468,4562,4797,5695,6251 plus 1040ES & 2210 update. Entry to one form is autoposted to other forms. IRS approv'd printouts. Calcs all taxes in 6 secs & cks for errors. Use for planning or actual '83 return. Sample return & detailed instr. 256K template. $65. CODs OK.

Austin Scientific, 1259 El Camino, #260ST
Menlo Park, CA 94025; (415) 364-0240

1040PLAN TEMPLATES $45

1983 Tax planning & preparation template for Lotus 1-2-3. Other calc spreadsheet programs are too small & slow to run 1040PLAN. Flexible, up to four different alternatives can be computed at once. Follows IRS forms, by line number, includes: form 1040, schedules A,B,C,D,E,G,W, & SE & forms 2119, 2441, 3468 & 6251. Req. 256K.

William A. Permar, CPA, 1125 Sunnyhills Rd.
Dpt. ST, Oakland, CA 94610
(800) 227-1617; In CA (800) 772-3545 exten. 644

FINANCIAL ANALYST PROGRAM

Perform many financial calculations with one easy-to-use program. Our program allows you to analyze investments, compute depreciation schedules and amortize loans. Price $50. Syntax Systems, 6642 S 193rd Place, Suite N107, Kent, WA 98032; (206) 251-8438.

GENERAL LEDGER + GRAPHICS

A user friendly **GL system integrating screen & printer graphics**. Graphs and reports may be selected by cost center or company. $125. (918) 663-3436 ; TSI, Box 55329-S, Tulsa, OK 74155.

POINT OF SALE

PC CASH REGISTER turns your PC into a cash register. FEATURES: inventory, sales reports, credit controls, commissions and customer base. OPTIONS: cash drawer, purchase orders, mail labels and invoicing. $395 & $495. Write or call (415) 927-0990. Data Consulting Group - DATA★EASY Software, 12 Skylark Dr. #18, Larkspur, CA 94939.

LANDLORD'S LOG

Just in time for your tax return. Use it now to prepare rental schedules for tax filing—use it later for monthly record keeping. Disk holds 30 properties (res. & apts.) with 100 tenants. Instant cash flow analysis-depreciation-summary reports. 2 drives. ONLY $89 (shp. & hand. incl.). Brochures available. Send check to: PRACTICAL SOFTWARE, 4162 Windsor Dr., Huntington Beach, CA 92649; (714) 842-0386.

NYSE STOCKS DATABASE

All common stocks listed on NY Exchange in dBase II file on 5 1/4" disk. Structure contains 3 fields: Wall St. Journal listing, full company name and symbol. Send $27 to: duWars, Box 6662, Greenville, SC 29606.

1984 ERGONOMIC COMPUTER FURNITURE — FREE CATALOG

Designs from over 20 national manufacturers
★ Discount Prices ★ Never a shipping charge
★ Visa - MC ★
Interior Design Systems, 3641-S St. Marys Place, NW, Washington, DC 20007.

THE PRICEBOOK

A multi-purpose system that outputs (to disk and/or printer) quantities, product descriptions, price extensions & totals. Ideal for sales & insurance agents, estimators, subcontractors, buyers, billing personnel et al. $99.95; Manual alone: $15. RJL Systems, 106 New Haven Ave., Milford, CT 06460; (203) 878-0376.

– *Professional* –

REAL ESTATE SOFTWARE

for IBM, APPLE, TRS-80 and CPM Computers

- PROPERTY MANAGEMENT SYSTEM: $525

Tenant Information	Owners Report
Late Rent Report	Operating Statement
Vacancy Report	Prints Checks
Expired Leases	Rent Receipts
Rent Statements	Consolidated Reports
Graphics	EASY TO USE

- PROPERTY LISTINGS/COMPARABLES: $425

—SCREEN BY→	Max/Min Price
	City/Memos/Codes
22 Items/Listing	Max Price/Income
Property Comparisons	Max Price/Sq. Foot
Homes or Income	Min Cashflow

- REAL ESTATE ANALYSIS MODULES: $50/Module

Loan Sales/Purchase	Loan Wrap Return
Home Purchase	Tax Deferred Exchange
Income Prop Analysis	APR Loan Analysis
Property Sales	Loan Amortization
Depreciation/ACRS	Construction Cost/Profit

- WORD PROCESSOR – WORD STAR: $295

Realty Software Company™

At Computer Stores Everywhere
or Order COD Direct
Cal Residents add 6½% Sales Tax
(213) 372-9419

Suite #229 1926 S. Pacific Coast Hwy., Redondo Beach, CA 90277

STATISTICS + GRAPHICS

A complete set of STAT functions + Curve Fitting + Multiple Regress + Integrated Business Graphics + SOURCE CODE $225. (918) 663-3436, TSI, Box 55329-S, Tulsa, OK 74155.

Education

TEST SCORING

At last! MULTIPLE CHOICE TEST SCORING the easy way with the use of your IBM PC. Will accept input from the keyboard or SCAN-TRON Optical Mark Readers. Reports include Item Analysis, Rank Order Listing, Statistical Report, Individual Score Reports and Class Results Report-$295. Contact Ormond Software, 218 Frontenac Rd., Marquette Hts., IL 61554; (309) 382-2335.

Graphics

PC ART DISK AND BOOK

Interested in using your PC for a nonfigurative geometric artistic experience? Req. Color Bd. & Mon. Try one of many by noted sculptor & educator Leroy Lamis. $50.

```
10 screen 1:CLS:KEY OFF:COLOR 5,0
20 for I=1 TO 118 STEP .7
30 circle (160+I*COS(I),100+I*SIN(I)),
I*5,RND*2+1,,,90/1:next I
```

PC Art, 3101 Oak St., Terre Haute, IN 47803.

PC-TITLE/PC-PROJECTOR

See PC mag's (Dec., Pg. 619) review of this useful tool for creating full color professional text for slides, screen displays & flyers. Let the PC, PCjr, XT or Compaq computer become your electronic slide projector for manual or time-controlled "slide shows". Add complete graphic capabilities with the much acclaimed PCcrayon. PC-Title/Projector $49.95. PCcrayon $44.95. Mention this ad and get all 3 programs for $79.95. VISA/MC. PC Resources, Inc., 620 Hobart Terrace, Santa Clara, CA 95051; (408) 243-4169.

INSTANT FLOWCHARTING

Create Edit & Print 4 by 8 figure **flowcharts & diagrams** + lines, arrows & text. $55.

TSI, (918) 663-3436
Box 55329-S, Tulsa, OK 74155.

Hardware

CURSOR_PROMPT. Design hardware cursors to indicate all SHIFT & TOGGLE states! DOS 1.1/2.0. Color/Mono. PC/XT & WORK-ALIKES. Used successfully with BASIC, WordStar, dBII, Multiplan, ProKey, etc. $10. TadAlex Software, 10834 Dixon Dr. S., Seattle, WA 98178.

SUPPLIES PRICES REDUCED!!

RIBBONS-Apple DMP/C.Itoh/Nec 8023 $4.29; Espon MX70/80 $3.85. STORAGE UNIT-75 capacity $15.75/10 capacity $1.49. For price list call (415) 778-2595 or write Argonaut Dist., 1104 Buchanan Rd. STI, Antioch, CA 94509.

VERBATIM DSDD DEMO DISK

★★★ state-of-the-art ★★★
Technical Stock Market Analysis Program
RUNS ON IBM/CLONES
Send $7 to:
OTI
6652 Chamois Circle
Cypress, CA 90630

Home

HOME MAILING PROGRAM

CHRISTMAS LIST is a personal mailing program. Prints any format pin-feed stock: labels, 3x5 cards, tags, etc. Maximum 3,000/disk. Sorts. Req.128K, DOS, Epson/IBM printer, 80-col. display.$30 to Johnson Software, 420 Pier Ave., Suite 195, Hermosa Beach, CA 90254.

COLLEGE FOOTBALL

Who is Number 1? Now you can play the Top Ten Teams against each other and determine the National Champion! You be the coach and play against a friend or the Computer or just sit back and watch. Created by an award-winning coach. Full Graphics. Requires IBM PC with at least one disk drive, color monitor and DOS 2.0. Send $29.95 to SoftJohn Co., 2627 Vinyard St., Huntsville, AL 35811.

THE RECIPE FILE was created by and for homemakers. Saves space, time, and money while you enjoy mastering your IBM PC. If you can use a recipe you can use this program. On disk $18.50. Send check or MO to Danne, div. of NADO Games, Box 752, Havertown, PA 19083.

GET CONTROL OF YOUR LIFE

Let *Fact-Finder* keep your notes, books, files, letters, clippings, phonograph record, slides and miscellaneous papers fully indexed for instant accessibility in home/office. You program nothing. Requires PC DOS 1.0, 1.1 or 2.0, 128K, one to four floppies or hard disks, monochrome or color display. $249. Satisfaction guaranteed or money back. Order *Fact-Finder*: Granite Software, Box 3024, Princeton, NJ 08540; (609) 896-2080.

Home Arcade

FLY YOUR MICROSOFT FLIGHT SIMULATOR WITH JOYSTICKS

Patch adds joystick control of elevators and ailerons. Send $12 for patch diskette to FSPATCH, 2311 10th Ave. NW, Rochester, MN 55901.

Home Education

HELP FOR PC USERS

The *only* HELP facility for the IBM/PC... thousands in use by gov't, schools, Fortune 1000. Makes learning/using DOS much easier. Provides on-screen reference, no need to dig thru the book. Any PC, XT, jr, or compatible with one DSDD drive. Any DOS release. *FREE color setting program!* $29.95 (check/ MO) RSI, 8723 Woodleigh Drive, Houston, TX 77083.

YES! OUI! ¡SÍ!

Play your way to language success in Spanish, French or English. Our versions of the classic Hangman game are stocked with hundreds of challenging words and phrases. Foreign versions allow you to choose either English or foreign language clues. Enhanced by humor, sound effects. One language, $15. Any two, $25. All three, $35. Req. 64K, 80-col. display. NORLAND SOFTWARE, 1014A W. Badger Rd., Madison, WI 53713.

Publications

PC-USERS BOOK SPECIALISTS

Comprehensive updated catalog *PC BookSource* shows how to get more from a PC or compatible
*Master Your Hardware & Software
*Make Better Business Decisions
*Communicate with Other Systems. And more!
ALL BOOKS RATED, GRADED & REVIEWED
Annual catalog subscription with updates: $5. MC/ Visa. Call (800) 84-BOOKS or (800) 842-6657. *PC-BookSource*, 48D Solar Prk, Pawlet, VT 05761.

DYNAMIC DUO RETURNS!

Two new disk magazines for the IBM PC: PC FIRING LINE (for programmers) and PC UNDERGROUND (for non-tech folk) are available now. Send a self-addressed stamped disk mailer for your free copies, or $8 and we will provide the disks and pay postage. (Or obtain copies from your Users Group!) ABComputing, Box 5503, North Hollywood, CA 91616-5503; (818) 509-9002.

Services

3M DISKETTES...$20.95

Box of 10 5.25" SS/DD/RH diskettes. DS/DD/RH $29.95. $2 shipping. Ship in 24 hr. Check/MC/ Visa. Cactus Computer, 3090 E. Palouse River Dr. Sw305, Moscow, ID 83843; (208) 882-8603.

SOFTWARE JUNKIE??

RENT today's most popular software: recreational, educational. Buy at *big* discount. Free brochure.
The Soft Source-R Inc.
Dept. K, Box 2931, Joliet, IL 60434

SOFTWARE!—SOFTWARE!

Save, SAVE, SAVE, SAVE
(219) 534-1012
HOOSIER SOFTWARE
Box 275, Goshen, IN 46526

$8 INCLUDING DISK ???

Yes! We have over 600 pcs famous programs for business, education, entertainment, etc., write immediately with $1.00 to Reliant, Box 33610, Sheungwan, Hong Kong.

TAPE/DISKETTE CONVERSION

Tape to diskette data conversion for IBM-PC. Half-inch, 9 track, 1600 BPI magnetic tape copied to IBM-PC 5 1/4" diskettes. Write or call C.M. Service Co., Box 1152, Shawnee Mission, KS 66222; (913) 384-2006.

3M SCOTCH DISKETTES $20.50

Authorized 3M distributor. Buy wholesale 5.25" SS/ DD $20.50. DS/DD $25.50. Prompt delivery! For price list call (415) 778-2595 or write Argonaut Dist., 1104 Buchanan Rd. STI, Antioch, CA 94509.

MAGAZINE ARTICLE INDEX

ASCII file—1700+ articles from all issues PC World, PC Age, PC Tech Journal, PC Magazine, Softalk & selections from Byte & DDJ. Retrieve using your program, EDLIN or PC-FILE. $9.95. Ampware, Box 30279, Cincinnati, OH 45230.

SUPER SOFTWARE SAVINGS

For a complete catalog of personal and small business computer software and hardware at excellent prices, write: SBCC, Box 1191, Thousand Oaks, CA 91360; (805) 492-9391.
Educational Software Specialists!

SAVE AT GOLEM COMPUTERS

Our ** SOFTWARE AND HARDWARE ** prices are lowest. All major brands are available. We carry business, education and entertainment software. Call for *FREE* catalog. (800) 345-8112. Pennsylvania (800) 662-2444.

IBM/PC USER NAMES

Over 30,000 names for rent @ $75/1000.
TSI
(918) 663-3436
Box 55329-S, Tulsa, OK 74155

TAPE/DISKETTE COPY SVC.

We'll copy your IBM formatted 9 track, 1600/800 bpi tapes to a single/double sided, 5 1/4" PC DOS formatted diskette. We can also copy IBM PC diskettes to CP/M 8" diskettes. Call Tracy Coker at *(619) 474-2010*, COMPUTER COGNITION, 225 W. 30th St., National City, CA 92050.

Strategy

SLEUTH

A murder has just been committed! Can you search the house, question the guests and find the murder weapon before the murderer becomes suspicious and disposes of you? Each game is different; changing floor plans, secret passageways, many possible murder weapons. Option to choose own friends as guests. $20. Requires 64K, 80-col display. NORLAND SOFTWARE, 1014A W. Badger Rd., Madison, WI 53713.

CLASSIFIED

Utility

MIC-ROSORT SORT/MERGE
Sorts fixed or variable records on up to 25 control fields (ascending or descending). Includes or excludes records on multiple and/or criteria. Handles multiple files using DOS 1.1 or 2.0; supports path names. Written in PC Assembler — FAST! Conversational screens, complete user manual. $75. Mic-roCode, 3600 Gayot, Anchorage, AK 99504.

FASTER, EASIER BASIC
Save 40% of your time. Eliminate line numbers. Name subroutines. Put library files in separate files. Use multi-line IF-THEN-ELSEs. Format output packed or pretty. The Exterminator for IBM compiler or interpreter. 64K/1D; 96K/2D; HARD DOS 1.0, 1.1, 2.0. Flyer. $39. Check/Visa/MC.
MicroMedia - Dept 90121
Box 33071, Denver, CO 80241

FREE FREE FREE
BASIC AIDS 2.0 FACT SHEETS and our guide "MAKE YOUR PC PROFITABLE" are sent free to persons who request them. This new release of BASIC AIDS is the most powerful program DEVELOPMENT/DOCUMENTATION tool available! Tulsa Computer Consortium, Box 707, Owasso, OK 74055; (918) 747-0151.

THE PROWRITER UTILITIES
Complete control of your C.Itoh 8510/1550 and NEC 8023. DOS 1.1 & 2.0 compatible for $44.95 + tax.
PROCNTRL - On-Line Keyboard Control of printer
PROEPSIM - Epson Graftrax Simulator (eg 1-2-3)
PROSCR - Text/Graphics Screen Dump, 3 sizes
PROSTALL - WordStar 3.2 & 3.3 Installation
PROSET - Menu Driven Printer setup and more
VISA/MC Accepted. Courtrin Enterprises, Box 231190, San Diego, CA 92123; (619) 569-8308.

DEBUG WITH BUGSCREEN
Symbolic debugger for IBM PC Macro Assembler. Windows for source, registers, memory, flags, stack. Dynamic display highlights changes. Scroll the listing. Alter registers, flags, memory. Display as decimal, character, hex. Single-step, set general breakpoints, execute procedures at normal speed. $95 (demo $7).
RDT Software, Box 96634, Weatherford, OK 73096.

ASYNC ADAPTER SUBSYSTEM
ASAP and ASAP-C (ASynchronous Adapter Package) are *interrupt-driven* device drivers for the IBM Async Adapter which interface to your assembly or Lattice C programs so async comm can be embedded in your application software. DK Softworks, Box 9802 #297, Austin, TX 78766.

SCREEN HANDLER UTILITY
Programmers' utility allows easy definition of screen formats. Save hours of coding time. No limit to the # of screens in a program. Color Screens. Menu-Driven. Uses PC-DOS BASICA. $129.95. User's Manual only $7.95. Marc-Ware, Inc., 2028 Buffalo Terrace, Houston, TX 77019; (713) 524-1295.

BASIC COMPACTOR
POSTBASE compacts & speeds-up IBM-PC BASICA programs. Compacted programs produce smaller ".EXE" files when compiled. POSTBASE has many features making it a valuable programming tool such as listing program variables. Req 128K and 80 columns. $59.95 (CA res 6%). Craftwork Engineering, Box 27449, San Diego, CA 92128.

PCBUS RUNS DATABUS(R)
IBM PC implementation of Datapoint's Databus (R) language. ISAM, string handling, arithmetic, more. Fast, compact, thousands of applications available. $750 complete; Reference manual, $30. Dealers inquire. Databus is a trademark of Datapoint Corp.
Infopoint Systems, Box 380275
Duncanville, TX 75138; (214) 780-1909.

MAG TAPE TO DISK CONVERSION
Any 9 track 1600 BPI Magnetic Tape converted to: *8 inch:* • CP/M • DISPLAYWR. • IBM SYS/?? *5 1/4 inch:* • *APPLE II* • ZENITH • TELEVIDEO *IBM PC* • KAYPRO • MORROW • XEROX SUPERBRAIN • ALTO • DEC VT180 • DEC RAINBOW TRS-80 • EPSON QX10 • EAGLE • *MANY OTHERS* Disk to disk and disk to tape conversion for a variety of other formats also available. Pivar Computing Services, Inc. 47 W. Dundee Rd., Wheeling, IL 60090; (312) 459-6010

FOR GEMINI-10X/15X USERS
Unleash the full power of your printer! FONTEDIT allows you to redefine any or all of your printer characters. Fast and easy, with impressive results. Includes sample "hi-tech" typeface that downloads from DOS. Mention this ad for a free "surprise" typeface — great for letters to friends. Complete documentation/tip book included. Editor requires BASIC. $29.95 from TRILOGIC DEVELOPMENT ASSOC., Suite 212, Box 3004, Corvallis, OR 97339.

Softalk/IBM's classified advertising section offers a considerably less expensive way than display advertising to reach tens of thousands of IBM Personal Computer owners.

Classified advertising space is available at the rate of $10 per line for the first ten lines, with a five-line minimum. Each line over ten lines is $25 per line. Ad copy should be received no later than the 10th of the second month prior to the cover date of the issue in which you want the ad to appear. Payment must accompany ad copy.

The publisher reserves the right to reject any advertising that he feels is not in keeping with the publication's standards.

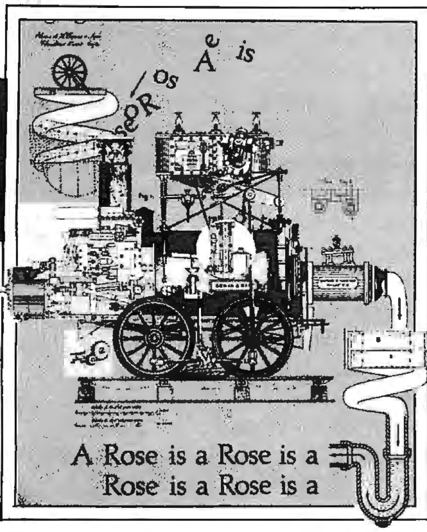
Heads will be set in 10-point boldface, all capitals only. Italics are available for body text only; please underline the portions you would like italicized.

The body text of the ad will hold roughly 45 characters per line. Spaces between words are counted as one character. Heads will hold roughly 24 characters per line, with spaces between words counted as one character. Please indicate whether you would like the head centered or run into the text.

Please write or call for additional information.

Softalk/IBM Classified Advertising
Box 7040
North Hollywood, California 91605
Attention: Linda McGuire Carter
(818) 980-5074

THE PROCESSED WORD



by *Terry Tinsley Datz and F. Lloyd Datz*

Wordvision and *Word Proof*

This year the well-dressed PC will be sporting keytops by Bruce & James. In an all-out effort to put first-timers at ease with *Wordvision*, Bruce & James Program Publishers has splashed the IBM's keyboard with shades of red, yellow, green, blue, and shocking pink. Instead of memorizing commands, you match the colors on your keys with *Wordvision*'s color-coded menus.

As simple as it is to learn, *Wordvision* isn't short on features. Its high points include a surprisingly powerful editor, on-screen text formatting, automatic paragraph reform, and, best of all, an ability to adapt to the way you like to work.

Overall Design. *Wordvision* has named the PC's inner row of function keys color keys; each wears a different color and symbol. Forget about F2 and F4 and think in terms of red club, yellow diamond, green heart, blue spade, and pink sun. Color keys do different things depending on where you are in the program; you keep up with their changing functions by matching them with the corresponding color and symbol in the menu at the bottom of the screen. If you have a monochrome display, you match just the symbol.

A more subdued set of keytops covers the control keys, the remaining five function keys, and the numeric keypad. Each keytop bears a descriptive name such as Format, File, Print, Undo, Adjust, or Stop (in addition to the PC's standard key legends). Unlike the colored function keys, these keys perform only one role apiece.

Installation. Most word processors assume you have a working knowledge of DOS; *Wordvision* assumes nothing. When you boot the master disk, a series of pictures and simple instructions helps you format a blank disk and tailor a working copy to your likes and needs. Pictures of your disk drives as well as the master and blank disks flash on and off to let you know which disk goes in which drive. To soothe jittery first-timers, *Wordvision* explains what it's doing every step of the way.

As a concession to those who like their keyboard unencumbered, the installation program thoughtfully asks whether you're going to use the keytops. If you are not, the menus are altered so that they display function key numbers instead of symbols and colors. The installation program then asks if you have a parallel or serial printer adapter; if you aren't sure, it checks to see. If you're connected to a serial printer, *Wordvision* automatically writes a batch file using the setup information you provide. It even takes care of writing an autoexec file if you want your working copy of the program to be self-loading.

Text Entry and Editing. *Wordvision*'s editing screen reserves two lines at the top for a status area and two at the bottom for menus, leaving twenty lines for text entry. The topmost line of the screen shows you the cursor's current direction and position (including column, line, and page counters), the percentage of memory your text occupies, and the status of the vision keys (more about these multipurpose keys later). On the far end of this line is a caps-lock indicator, and in the center is an inverse-video bar that looks like a gas gauge. As you scroll through your file, a needle inside the bar moves from left to right tracking your relative position. Below the status line is a ruler that expands and contracts as you change your margins.

The lines at the bottom of the screen display the name of the current menu and the assignments of the five color keys in that menu. As you edit, the emphasis and excerpt marking menu is generally in effect, with the color keys assigned to underlining, boldfacing, marking blocks, and inserting hyphens.

The cursor is the editing screen's most unusual feature. A small, unblinking square, it looks ordinary enough when it rests at the home position. Move it through your text, though, and you'll find that it never alights under a character or space. Instead, it creates its own phantom spaces by wedging itself between characters. Why this unusual approach? Presumably, it eliminates confusion as to exactly where your next command (deleting, for example) will take place. However, it has the disconcerting effect of causing your text to ripple as phantom spaces open and close when the cursor moves from line to line.

However, *Wordvision* leaves little to complain about in options for moving the cursor. The arrow keys move the cursor by character and by line, and the home and end keys make the cursor jump to the top and bottom of the screen and document. You can also move both forward and backward by word, sentence, and paragraph using the left and right vision keys (the num-lock and scroll-lock keys in disguise).

Actually, the vision keys perform multiple tasks; besides moving the cursor, they also delete and transpose text. You activate them as cursor keys and set their direction by tapping the left or right arrow key. By themselves the vision keys move the cursor by word and line; combined with the shift key, they move by sentence and paragraph. Should you get confused, you can always refer to the status line at the top of the screen.

Wordvision refers to the page-up and page-down keys as *prev* and *next*: Logically enough, prev takes you to a previous screen while next jumps ahead. By combining prev and next with the shift key, you can browse through your file page by page. Whether you're scrolling by page or by screen, however, *Wordvision* holds you to a leisurely pace; it's not smart enough to skip screens or pages when you hit prev or next several times.

If you're working with wide documents, you can extend your field of view beyond the eightieth column by scrolling sideways to a maximum of one hundred thirty columns.

The editor is always in insert mode. What, no typeover mode? In order to protect against typing over valuable text, Bruce & James left it out; so, to correct typos, you have to erase first, then retype. As a consolation, however, *Wordvision* provides a swap mode, which you call with the PC's insert key (renamed the swap key). By hitting the swap key when the cursor is positioned between any two letters, you can have those letters transposed. In addition, by combining the swap key with the vision keys, you can reverse the order of words, sentences, or even paragraphs.

The PC's delete key doesn't delete at all; instead, *Wordvision* uses it as a caps-reverse key to change the case of existing text. Depending on whether you press it alone or with the shift key, it moves the cursor forward or backward, changing lowercase letters to uppercase, and vice versa, as it goes. The vision keys extend the range of this key to work on entire words, lines, sentences, or paragraphs. For example, let's say you discover you've been typing along with the caps-lock key engaged, which can happen when you're typing fast and furious from printed copy. For a quick repair job, just use the caps-reverse key combined with the vision key to rescue your text a paragraph at a time.

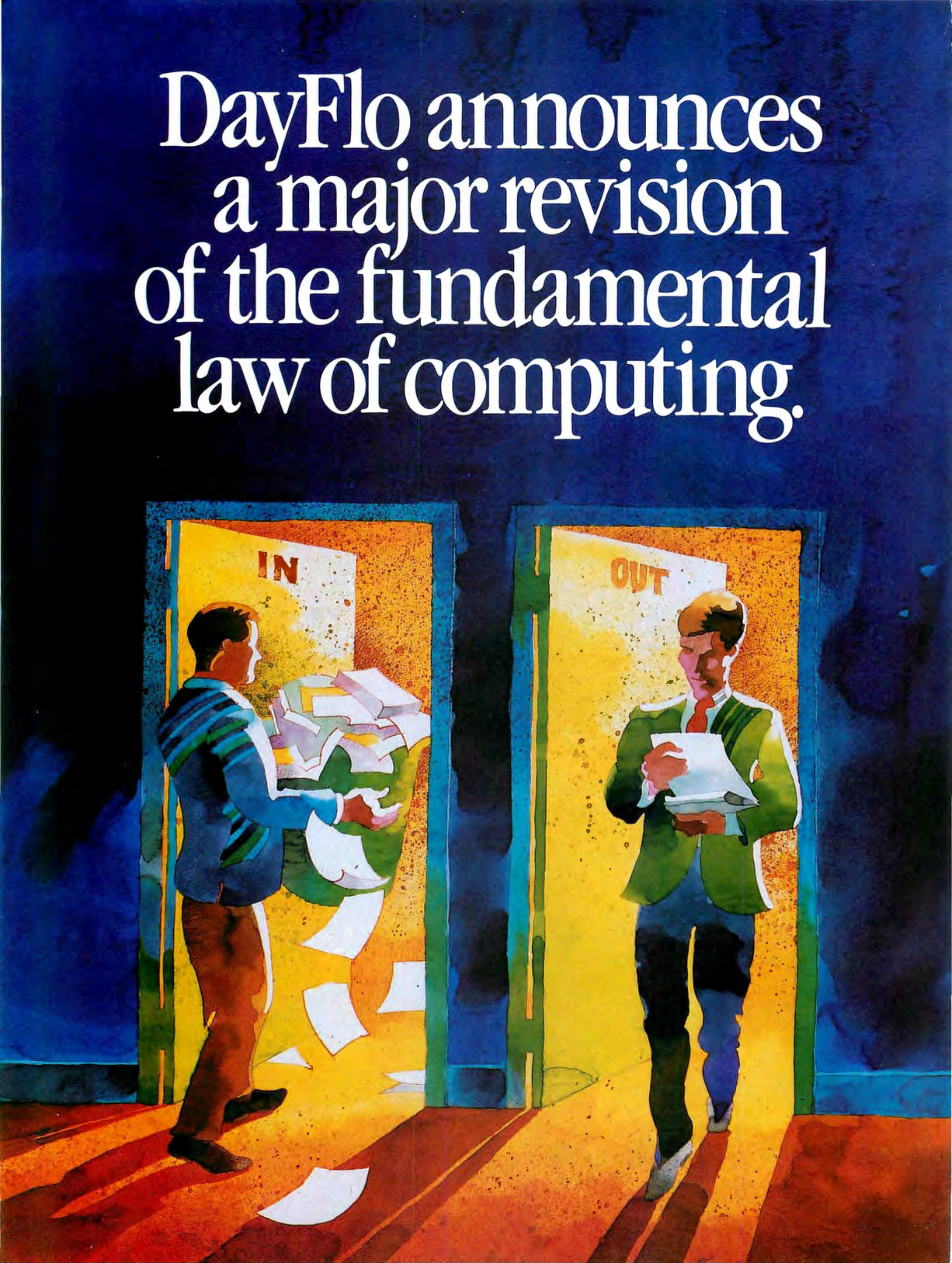
The backspace key deletes in both directions: Unshifted, it deletes to the left of the cursor; shifted, it deletes to the right. To expand the range of this key and delete entire words, lines, sentences, or paragraphs, you use your old friends the vision keys. Just press backspace first to designate them as deleting keys.

Thanks to *Wordvision*'s undo key (the minus key in the numeric keypad), no deletion need be permanent—assuming that you haven't touched any other keys in the meantime, that is. Hitting the undo key restores your text a character at a time; pressing it with the shift key brings what you deleted back in one lump. The undo key can work wonders restoring whole files that you've deleted by accident.

Wordvision handles block operations admirably. To define a block of text, you position the cursor at one end and tap the green heart key (F6), then move to the other end of the block and hit the green heart again. Once marked, the block (which *Wordvision* calls an excerpt) appears in inverse video. When you press the excerpts key (F1), *Wordvision* displays your options (erasing, moving, copying, saving to a separate file, or printing) and shows the marked text in a window at the bottom of the screen.

Most word processors have strict rules against defining several blocks of text at the same time; *Wordvision* does not. Although the most recently defined block automatically appears in the window, you can gain access to any others that you've marked by scrolling with the prev and next keys.

No matter how much revising you do, whether by adding or deleting a few words or


DayFlo announces a major revision of the fundamental law of computing.
IN
OUT

by massive cut-and-paste operations, *Wordvision* closes up gaps and keeps everything neatly aligned between the margins. This process seems so natural that, unless you're used to a word processor that makes you reform your text manually, you'll probably take it for granted.

Wordvision's search-and-replace functions give you a nice array of options. You choose whether you want partial matches as opposed to whole words and whether you want capitalization matched or ignored. If the word you're replacing is in boldface or underlined, *Wordvision* automatically emphasizes the new text to match. You can confirm replacements one by one, replace words globally without intervention, or limit the replacements to marked excerpts. The one thing you can't do is search backward from the cursor position.

In a limited sort of way, *Wordvision* allows you to set up macros for frequently used text. The catch is that you can define only five macros per file (one for each color key) and each macro can be no more than forty characters long. While forty characters may be ample for company names and most return addresses, it won't do you much good if you want to insert whole paragraphs. You can, however, save such paragraphs as separate files and merge them into your text at the cursor position.

You assign text to a macro by calling up the quick phrases control panel (a control panel is a menu with which you customize the program). The quick phrases panel displays the symbols for the five color keys, leaving space for you to type in your text after each one. Once you've assigned text to one or more of the color keys, you can recall that text by pressing *phrase* (the PC's alt key) followed by the appropriate color key.

Just like many housekeepers, *Wordvision* does windows grudgingly. To look at one part of your file while editing another part, you place a marker in the text you want to view. Then, when you press the windows key, *Wordvision* displays the first seven lines of the marked text in a window at the bottom of the screen. But that's it. You can't change the size of your window, scroll its text, or edit it—let alone look at a different file. So much for windows.

Formatting and Printing. *Wordvision* comes about as close as any word processor to displaying your work as it'll appear when printed. It shows full justification, double spacing, and even underscoring and boldfacing. However, *Wordvision* simply doesn't display page breaks. The only way of knowing where breaks will occur is to scroll through your file by page and observe where the cursor comes to rest (it lands at the top of each page). Let's hope you don't find any breaks in awkward places—such as just after a heading or in the middle of a table—because *Wordvision* doesn't allow you to force a new page. Nor can you protect tables and charts from being split between two pages since there's no way to insert a conditional page break.

You indicate how you want your document formatted by adjusting the values in the main control panel or the page appearance panel. The main control panel, which you call when you first enter the program, sets the options that affect your whole document, such as justification and line spacing.

To fine-tune your page layout, you use the page appearance panel. The right side of this screen lists current settings, including those for paper width and length, pitch and line height, and text width and margins. All except pitch and line height are expressed in inches, not columns. You use the arrow keys to move up and down the list; the prev and next keys vary each setting in half-inch increments.

The left side of the page appearance panel depicts a sheet of paper with its text and margin areas blocked out to conform to the current settings. As you change the settings in the list on the right-hand side, the picture changes to reflect the new values. At the same time, any related settings in the list also change. For example, if you increase text width by an inch, the text area in the depicted sheet of paper expands; at the same time, the left- and right-margin values in the list each decrease by half an inch. Because all the values from the page appearance panel stay with your document, you don't have to reset them on future edits.

Wordvision also allows you to make line-by-line adjustments to your format. For example, you can have lines or blocks of text centered, justified, or aligned flush against the right margin. You can't, however, vary line spacing—something you might want to do to set off quotations or indented paragraphs; you're stuck with whatever spacing you chose in the main control panel.

Although *Wordvision* has no hyphen-help feature, you can insert discretionary hyphens in multisyllabic words; these hyphens become active only if the word spills over at the end of a line. Since discretionary hyphens are invisible unless they're needed, finding and deleting one that's inactive can be a little tricky.

Anyone who works with outlines will appreciate *Wordvision*'s smooth handling of temporary margins. You set up temporary indents for either the left or right margin (or both) by hitting F5 (format), which activates a tab pointer in the ruler line. You move the pointer along the ruler and press the green heart key wherever you want to set an indent stop. Then, when you're typing a paragraph that you want indented, just press indent (the PC's control key); the temporary margin will stay in effect until the end of the paragraph. The same procedure holds for setting up tab stops, which can be either regular or decimal.

THE WORLD'S LARGEST COMPUTER MAIL ORDER FIRM

CONROY-LA POINTE™

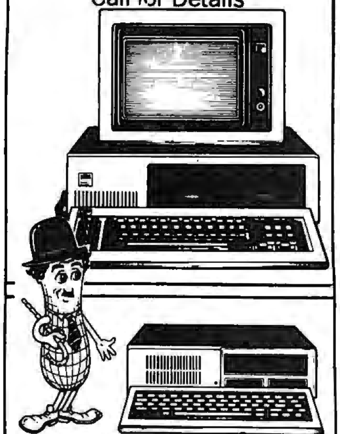
Formerly **Computer Exchange**

LOW PRICES TO PROFESSIONALS WHO KNOW WHAT THEY WANT AND KNOW HOW TO USE IT!

ALL MAIL: Conroy-LaPointe, Inc. 12060 Garden Place, Portland, OR 97223 SHOWROOMS AT: PORTLAND, OR or SEATTLE, WA — BOTH OPEN M-SAT 10-6

256K IBM®-PC or XT

320/360K Disk Drives by CDC
90 Day Warranty By Us
Call for Details



Coming soon——products for the PC Jr.

® 1984, Service Mark of Conroy-LaPointe, Inc

for the IBM-PC or XT DRIVES AND ACCESSORIES

CONTROL DATA

$219
$229 For One.
Same as now installed by IBM

320K/360K DS/DD DISK DRIVES
With Detailed Installation Instructions
30 Day Warranty by Factory Authorized Distributor
HALF $199 HEIGHT

PRINTER COMBO SALE
ONE TIME, QUANTITY LIMITED
OKIDATA ML80

80 cps,
Pin feed, 80 col.,
132 col. conden.
96ASCII, Graphics, Parallel
List Price $299

COMBO A $549 save $450
ML80 + dBase II (Apple or IBM)

COMBO B $494 save $500
ML80 + WordStar + Mail Merge + SpellStar + Star Index (Apple or IBM)

COMBO C $474 save $520
ML80 + Applicard (CP/M + 13 Features) + WordStar or InfoStar (Apple II+/e)

COMBO D $397 save $550
ML80 + 50 Generik™ DS/DD diskettes + Bank Street Writer + Home Accountant Plus + 3 Insoft Trix educational games. (IBM-PC)

COMBO E $291 save $420
ML80 + 50 Generik™ SS/SD diskettes + Bank Street Writer + Insoft 3 game pak + Home Accountant (Apple II +/e)

OUR AD #14

NATIONAL ORDER DESK **TOLL FREE (800) 547-1289**
Order Desk Hours: 6AM to 6PM PST
Oregon TOLL FREE (800) 451-5151 Portland: 620-9877
Hot Line For Information On Your Order (503) 620-9878

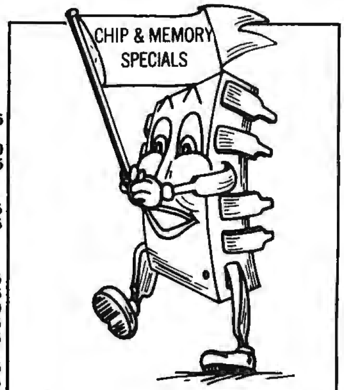
FREE GIFT COUPON
Use of our order forms qualifies you for a free gift with your order. Get on our mailing list now for order forms, and our new newsletter and sales specials announcement. Our customers are already on our list.

MAIL TO:
12060 Garden Place, Portland, OR 97223

NAME ____________________
ADDRESS ____________________
CITY ____________ STATE ____ ZIP ______

SUPPLY CENTER for IBM®-PC or XT

		LIST PRICE	OUR PRICE
AMDEK	MAI 4-in-1 Multiple Board, Color Graphics, Mono, 128K	$ 599	$ 519
AST	ComboPlus, 64K S/P/C	$ 395	$ 279
	ComboPlus, 256K S/P/C	$ 695	$ 495
	MegaPlus II, 64K, 2S/P/C	$ 495	$ 375
	MegaPlus II, 256K, 2S/P/C	$ 795	$ 595
	256K MegaPlus II Expander	$ 395	$ 295
	SixPakPlus, 64K, S/P/C +S/W	$ 395	$ 295
	SixPakPlus, 256K S/P/C +S/W	$ 695	$ 495
	SixPakPlus, 384K S/P/C +S/W	$ 895	$ 595
	For SixPak w/ Game Port, add	$ 50	$ 39
	I/O Plus II, S/P/C	$ 215	$ 150
	I/O Plus II, S/P/C/G	$ 265	$ 185
CCS	SuperVision, 132 col., mono. board	$ 799	$ 599
	Z Plus 64, fast Z80B, 64K para port	$ 875	$ 695
Chalkboard,	Power Pad, Req. Kit	$ 100	$ 73
*ComX	256K RAM Card with Fastrak™ RAM disk emulator and spooler software.	$ 495	$ 325
CURTIS	UNI-I, Monitor tilt & swivel base	$ 50	$ 39
	3 to 9 foot keyboard cable	$ 40	$ 30
	Vertical CPU "System Stand"	$ 25	$ 19
	Monochrome Ext. Cable Pair	$ 50	$ 35
HERCULES.	Monochrome Board	$ 499	$ 349
Key Tronic	KB5150 Std. keyboard	$ 209	$ 159
	KB5151 Std. keyboard NEW	$ 255	$ 209
Koala	Koala Pad™ w/PC Design	$ 150	$ 109
	Programmer's Guide	$ 15	$ 12
MAYNARD	Multifunction (6) Card, MFC	$ 89	$ 79
	Memory Card no RAM	$ 230	$ 169
SANDSTAR	Memory Card 256K	$ 499	$ 395
	Modules for Sandstar in stock		Call
MICROSOFT	RAMCard 256K	$ 550	$ 385
	SystemCard 256K	$ 625	$ 469
	SystemCard 64K	$ 395	$ 295
	Mouse	$ 195	$ 145
MOUSE SYSTEMS,	PC Mouse w/software	$ 295	$ 195
ORCHID	PCnet™ Starter Kit, LAN	$1490	$1190
	PCnet™ Circuit Board Kit	$ 695	$ 545
PLANTRONICS	Color Board & Colormagic, 16 color, w/Para	$ 559	$ 395
	Color Board & Draftsman, 16 color, w/Para	$ 559	$ 395
QUADRAM	★ Quadlink NEWEST VERSION	$ 680	$ 485
	Quadboard, no RAM, expand to 384K	$ 295	$ 215
	Quadboard 64K, expand to 384K	$ 395	$ 279
	Quadboard 256K, expand to 384K	$ 675	$ 525
	★ Quadboard, 384K	$ 795	$ 625
	Quadboard II, no RAM, expand to 256K	Call	Call
	Quadboard II, 64K, expand to 256K	$ 395	$ 285
	Quadboard II, 256K, 6 function	$ 595	$ 395
	Quad 512 × 64K plus serial port	$ 325	$ 265
	Quad 512 × 256K plus serial port	$ 550	$ 420
	Quad 512 × 512K plus serial port	$ 895	$ 625
	Quadcolor I, board, 16 colors	$ 295	$ 225
	★ Quadcolor II, board, use with Quadcolor I	$ 275	$ 209
	★ Quadchrome, 12" RGB Monitor	$ 795	$ 499
	Quadscreen, 17" 968 × 512 Monitor	$1995	$1595
Tecmar	1st MATE, 64K	$ 389	$ 295
	1st MATE, 256K	$ 589	$ 439
	Captain, 64K S/P/C/Prog	$ 424	$ 324
	Captain, 384K S/P/C/Prog	$ 795	$ 595
	Wave, 256K (short brd.)	$ 499	$ 369
	Bosun, S/P/C (short brd.)	$195	$145
	Graphics Master	$ 695	$ 575
Titan	Accelerator PC (8086 + 128K)	$ 995	$ 750
TG PRODUCTS.	Joystick	$ 60	$ 40
WICO,	IBM-PC Mouse	$ 100	$ 69



Prices and availability subject to change. Call.

$55
★ 9 Each, 4164, 200 ns, MEMORY CHIP KIT
90 Day Warranty by us

$325 $295 Two or more.
★ **ComX 256K RAM BOARD**
Fully Compatible 1 Year Limited Warranty by ComX
With Fastrak RAM Disk Emulator and Spooler Software
Works on DOS 1.1, 2.0 or 2.1

★ MEANS A BEST BUY

SOFTWARE for IBM®-PC or XT

BUSINESS

	LIST PRICE	OUR PRICE
ALPHA, Database Manager II	$ 295	$ 185
ASHTON-TATE		
★ dBase II, req. PC-DOS & 128K	$ 700	$ 385
dBase II User's Guide (Book)	$ 30	$ 20
Everyman's DB Primer (Book)	$ 15	$ 12
The Financial Planner	$ 700	$ 395
Friday	$ 295	$ 199
APPLIED SOFT. TECH., Versaform	$ 389	$ 265
ASK MICRO, GL,AR,AP,INV or PR, each	$ 495	$ 295
BRODERBUND, Bank Street Writer	$ 80	$ 56
BPI, Gen'l Acctg,AR,AP or PR, each	$ 595	$ 395
CHANG LABS, Micro Plan	$ 495	$ 335
★ **CONTINENTAL**, Home Accountant	$ 150	$ 89
Tax Advantage	$ 70	$ 45
FCM (Filing, Cataloging, Mailing)	$ 125	$ 89
Property Management	$ 495	$ 329
DOW JONES, Market Analyzer	$ 350	$ 279
Market Manager	$ 300	$ 239
Market Microscope	$ 700	$ 525
FOX & GELLER,		
Quickcode, dGraph, Grafox or Oz, each	$ 295	$ 195
dUtil (MSDOS or CP/M86 each)	$ 99	$ 59
HAYDEN, IBM Pie Writer	$ 200	$ 135
Pie Speller or Sargon III, each	$ 50	$ 34
HOWARDSOFT,		
Tax Preparer, 1984-for 1983 year	$ 295	$ 220
HUMAN EDGE, Management or Sales, ea.	$ 250	$ 185
IUS, EasyWriter II System	$ 350	$ 259
EasySpeller II	$ 225	$ 149
Business System: GL+AR+AP	$1495	$ 995
GL,AR,AP,OE or INV, each	$ 595	$ 395
★ **INSOFT**, Data Design (easy to use DBMS)	$ 225	$ 169
GraFORTH (animated 3D graphics)	$ 125	$ 95
LIFETREE, Volkswriter	$ 285	$ 195
★ **LOTUS**, 1-2-3	$ 495	$ 329
QUE, Using 1-2-3 (Book)	$ 15	$ 12
MICRO LAB, Tax Manager for 1983	$ 250	$ 169
MICROPRO, WordStar®	$ 495	$ 239
MailMerge™	$ 250	$ 129
SpellStar™	$ 250	$ 129
★ WordStar Professional, 4 Pak	$ 695	$ 395
Options Pak, SS/MM/SI	$ 295	$ 175
StarIndex™	$ 195	$ 109
InfoStar™	$ 495	$ 259
★ **MICRORIM**, R:base, Series 4000	$ 495	$ 335
MICROSOFT, Multiplan	$ 250	$ 169
Word	$ 375	$ 259
Word with Mouse	$ 475	$ 325

BUSINESS

	LIST PRICE	OUR PRICE
MICROSOFT, Financial Statement	$ 100	$ 69
Budget	$ 150	$ 99
MONOGRAM, Dollars & Sense	$ 165	$ 110
OPEN SYS, GL,AR,AP,PR,INV or PO,each	$ 695	$ 429
★ **OSBORNE/COMX**, (Book & Business, Statistics & Math Programs on DS/DD Disks)		
Some Common Basic Programs (70 ea.)	$ 100	$ 69
Practical Basic Programs (40 each)	$ 100	$ 69
PBL, Personal Investor 1.1	$ 145	$ 99
PEACHTREE, Peach Pak (GL,AR&AP)	$ 395	$ 239
Peach Text 5000	$ 395	$ 239
PEARLSOFT, Personal Pearl (DBMS & MIS)	$ 295	$ 195
★ **PERFECT**, Perfect Writer™	$ 349	$ 219
Writer & Speller, 2 Pak	$ 399	$ 249
Perfect Filer™ or Perfect Calc, each	$ 299	$ 149
Perfect Writer, Speller, Filer, Calc (4)	$ 969	$ 499
SATELLITE, Word Perfect	$ 495	$ 255
SOFTWARE ARTS, TK! Solver	$ 399	$ 299
SOFTWARE PUBLISHING, PFS:File	$ 140	$ 94
PFS:Report	$ 125	$ 84
PFS:Write	$ 140	$ 95
PFS:Graph	$ 140	$ 95
SOFTWORD SYSTEM, Multimate	$ 495	$ 395
SORCIM, SuperCalc 2	$ 295	$ 195
SuperCalc 3	$ 395	$ 265
SSI/SATELLITE, WordPerfect	$ 495	$ 375
Personal WordPerfect	$ 195	$ 149
STC/SOFTEC, The Creator	$ 300	$ 195
STONEWARE, Advanced D.B. Master	$ 595	$ 395
SYNAPSE, File Manager	$ 100	$ 67
SYNERGISTIC, Data Reporter	$ 250	$ 169
T/MAKER, T/Maker III	$ 275	$ 169
VISICORP, VisiCalc IV	$ 250	$ 179
VisiFile or VisiSchedule	$ 300	$ 219
Desktop Plan I	$ 300	$ 219
VisiWord with VisiSpell (128K)	$ 375	$ 269

UTILITY & SYSTEM

1983 CL SOFTWARE AWARD:
"Copy II PC by Central Point Software is still the best software buy of 1984. It will copy more copy protected software and faster than any other backup system. Unlike other copiers it makes an exact duplicate of your original and it does 100% verification of copy. Documentation is excellent."

	LIST PRICE	OUR PRICE
★ **CENTRAL POINT**, Copy II PC Backup	$ 40	$ 30
★ **COMX**, Fastrak™, RAM/Disk emulator and printer spooler program. Works on any PC/DOS version or RAMCard. Menu Driven	$ 100	$ 59

UTILITY & SYSTEM

	LIST PRICE	OUR PRICE
DIGITAL RESEARCH,		
Concurrent CP/M-86™	$350	$225
Concurrent CP/M-86™ w/ windows		Call
CP/M-86™	$ 60	$ 40
CBASIC 86™	$ 200	$ 135
CBASIC Compiler (86 or MSDOS), each	$ 600	$ 365
Pascal/MT+ (CP/M-86)	$ 400	$ 269
Pascal/MT+ (MSDOS)	$ 600	$ 399
PL/1 (MSDOS) or CP/M-86, each	$ 750	$ 499
Access Mngr. (MSDOS or CP/M-86), each	$ 400	$ 179
Display Mngr (MSDOS or CP/M-86), each	$ 500	$ 339
Speed Prog. Pkg. (CP/M-86)	$ 200	$ 135
CIS COBOL-86	$ 850	$ 525
DR LOGO-86	$ 100	$ 69
HAYES, Smartcom II (Data Com.)	$ 119	$ 89
INSOFT, GraFORTH (animated 3D graph.)	$ 125	$ 95
MICROSTUF, Crosstalk XVI (Data Com.)	$ 195	$ 129
MICROSOFT, muMath/muSimp	$ 300	$ 199
Business BASIC Comp.	$ 600	$ 399
Pascal Compiler	$ 350	$ 250
C Compiler	$ 500	$ 339
BASIC Compiler	$ 395	$ 269
FORTRAN Compiler	$ 350	$ 250
COBOL Compiler	$ 750	$ 495
NORTON, Utilities 2.0, 14 programs	$ 80	$ 65
ROSESOFT, Prokey	$ 75	$ 50

HOME & EDUCATIONAL

	LIST PRICE	OUR PRICE
Atari, Centipede, PacMan or Donkey K.,ea.	$ 35	$ 28
Beagle Bros., Beagle Bag	$ 30	$ 22
Bluechip, Millionaire	$ 60	$ 40
Broderbund, Choplifter or Lode Runner, each	$ 35	$ 25
Arcade Machine	$ 60	$ 40
Apple Panic	$ 30	$ 21
BudgeCo., Pinball Constr. Set	$ 40	$ 27
★ **Continental**, Home Accountant	$ 75	$ 49
Datasoft, Aztec or Zaxxon, each	$ 40	$ 27
Davidson, Math Blaster!	$ 50	$ 34
Edu-Ware, (Large Inventory)		Call
Hayden, Sargon II (Chess)	$ 35	$ 29
Sargon III (Chess)	$ 50	$ 34
Infocom, Zork I,II,III, or Starcross, each	$ 40	$ 27
Koala, Full line in stock, CALL		Call
Learning Co., (Large Inventory)		Call
Micro Lab, Miner 2049er	$ 40	$ 27
Microsoft, Typing Tutor	$ 25	$ 17
Monogram, Dollars and Cents	$ 100	$ 69
Origin, Ultima III	$ 60	$ 40
Scarborough/Lightning, Mastertype	$ 40	$ 27
Sierra/On-Line, Ultima II	$ 60	$ 40
Sir-Tech, Wizardry	$ 50	$ 39
Spinnaker, Kindercomp (others in stock)	$ 30	$ 20
Sub Logic, Flight Simulator II	$ 50	$ 37

ORDERING INFORMATION AND TERMS: MAIL TO: 12060 Garden Place, Portland, OR 97223 Include telephone number and double check your figures for SI&H. All items usually in stock. Cashiers Checks, Money Orders, Fortune 1000 Checks and Government Checks, we immediately honor. Personal or other Company Checks allow 20 days to clear. No C.O.D. Prices reflect a 3% cash discount so ADD 3% to above prices for VISA or MC. For U.S. Mainland, add 3% ($5 minimum) for shipping, insurance and handling (SI&H) by UPS. UPS ground is standard so add 3% ($10 minimum) more for UPS Blue for SI&H. Add 12% total ($15 minimum) for SI&H for US Postal, APO or FPO. For Hawaii, Alaska and Canada, UPS is in some areas only, all others are Postal so call, write, or specify Postal. Foreign orders except Canada for S,I&H add 18% or $25 minimum for SI&H except for monitors add 30% or $50 minimum for SI&H. Prices subject to change, typo errors, and availability so call to verify. All goods are new, include warranty and are guaranteed to work. Due to our low prices and our assurance that you will get new unused products, ALL SALES ARE FINAL. Call before returning goods for repair or replacement. Orders received with insufficient SI&H charges will be refunded. ORDER DESK HOURS 6 A.M. to 6 P.M. PST, Monday through Friday and 10 to 4 Saturday. 6 A.M. here is 9 A.M. in New York

OUR REFERENCES: We have been in computers and electronics since 1958, a computer dealer since 1978 and in computer mail order since 1980. Banks: 1st Interstate Bank, (503) 643-4678. We belong to the Chamber of Commerce (503) 644-0123, Better Business Bureau and Direct Marketing Association, or call Dunn and Bradstreet if you are a subscriber. Fastrak™ and Generik™ are trademarks of ComX Corporation

Wordvision is stingy with headers and footers, restricting them to just forty characters. Although you can position them at the left, right, or center of either the top or bottom margin, once you've set them, you can't vary their position or content. You're out of luck if you want to print different headers on odd- and even-numbered pages.

To print your work, you call the print control panel, which gives you options such as printing a partial file, printing multiple copies, and setting the initial page number to something other than 1. Once the print panel values are set, you press the print key (F9) and indicate whether you want to print everything in memory, just what's on the screen, or marked excerpts. *Wordvision* doesn't, however, allow you to print in the background while you edit.

File Handling. *Wordvision's* file directory is like no other word processor's. In a radical detour from the traditional laundry list, it shows you a stack of overlapping file folders. Displayed at the top of each folder is a filename (which can be a generous forty characters long), the file's length in pages, and the time and date of its creation or last revision. Only five folders will fit on the directory screen at a time; if you have more files on the disk, you scroll through them five at a time. The top of the directory screen tells you which disk drive is active, the number of *Wordvision* files on the disk, and the percentage of disk space filled.

Wordvision's generosity with the length of filenames is a giveaway that it doesn't save your text as standard DOS files. However, by choosing *DOS Editor* mode from the main control panel, you can use *Wordvision* as a program editor or with files created by other programs. By switching back and forth between modes, you can convert documents created in normal *Wordvision* word processing mode to DOS files and vice versa, which means that you can use your *Wordvision* files with other programs and copy other programs' files into *Wordvision*.

Other convenient utilities allow you to copy and delete files and get immediate updates of how much disk space your files occupy. In the event that you try to save an oversized file to an overcrowded disk, you get a "sorry" message that politely suggests you use another disk. *Wordvision* has no objection to your saving two files under the same name, a handy option when you want to save several versions of the same manuscript. You can tell them apart by the time and date statistics on their file folder.

Wordvision has an automatic backup system that's ideal for absent-minded folks who seldom remember to save their work. At specified intervals (say, every twenty minutes) the program sounds a chime, then flashes a bulletin reminding you to save. Even better, you can ask *Wordvision* to automatically stop and save at any interval you want.

Wordvision offers still other options for customizing the program to the way you work. The keyboard control panel adapts your keyboard to suit the eccentricities of your typing abilities and needs; for example, you can choose how fast the keys autorepeat—from fast to not at all. You can also modify the workings of the caps-lock key by having it affect numbers and punctuation in addition to letters (as it does on typewriters)—or just numbers.

The screen panel controls similar details pertaining to the way the program displays your text. You can have all the blank spaces in your text filled with dots, for example, a handy feature for formats such as tables, where spacing is crucial. An option that will be appreciated by programmers allows you to modify the line counter in the editing screen so it counts by total lines rather than on a lines-per-page basis.

Wordvision doesn't do automatic disk buffering, so maximum file size depends on how much memory your system has. With the minimum requirement of 96K, you have to limit your files to eight double-spaced pages. This limit is all the more confining since you can't chain files together at print time. If you have 256K, *Wordvision* gives you a little more breathing room, allowing files to contain up to fifty pages.

Documentation and Support. You'll find *Wordvision's* manual a pleasure to use. Its one hundred fifty-six pages include color-coded examples and sample screens and both an index and table of contents. To make for easy reading while you're at the keyboard, it converts into an easel on which the pages open from top to bottom instead of from left to right. As you turn the pages, the top page is a tutorial that gives you step-by-step instructions, while the bottom page contains reference material about the same feature. This saves you from having to flip between a reference and tutorial section when you want information about a command.

Taking an honest approach, the manual tells you what *Wordvision* won't do (it won't work with print spoolers that rely on software or with keyboard modification programs, for example). There's also a listing of "sorry" messages that you're likely to encounter, complete with suggestions for recovery and where to find related reference material. For situations that may be confusing even though no error message is displayed, there's a "What if?" section. Unfortunately, the manual consistently refers to keys by their *Wordvision* names (green heart, for instance, instead of F6). That's fine if you're using the keytops but irritating if you're working at a naked keyboard.

For questions that the manual doesn't answer, Bruce & James runs a toll-free hot line.

The program disk isn't copy-protected.

Ease of Learning and Use. With its color-coded keys, simple menus, and down-to-earth instructions, *Wordvision* couldn't be easier to learn. In most cases you can tell what to do next by reading the keyboard and the screen. For exceptional cases, the tutorial or reference material in the manual always comes to your rescue. There's also an extensive network of help screens, but they're hard to get to and not all that helpful.

But if you're interested in maximizing your productivity, *Wordvision* may not be your best bet. Since the function keys do different things in different menus, it's virtually impossible to memorize what key does what. Instead, you have to wait for a menu to appear, then check to see the new key assignments.

Summary. At $79.95, *Wordvision* gives you more features per dollar than any other PC word processor. If you want to get into word processing with a minimum investment of time and money, here's your chance.

System Requirements. *Wordvision* requires a minimum of 96K. Although it works fine with a single disk drive, two are ideal; you can also use a hard disk. A word of warning about using DOS 2.0, however: Though *Wordvision* is compatible with DOS 2.0, it doesn't support

hierarchical directories. The people at Bruce & James say a PC*jr* version is in the works.

The program disk comes with a limited number of printer drivers—currently only the IBM and Epson dot-matrixes, the Diablo 630, and the NEC 3550.

Word Proof

Another program that stretches your word processing dollar is *Word Proof*, brought to you by none other than Big Blue. *Word Proof* gives you a spelling checker, thesaurus, mini-word processor and anagram solver—all for only $60. If IBM doesn't realize that *Word Proof* is worth a lot more than the cost of a DOS update, let's not tell them.

Overall Design. Although *Word Proof* includes a rudimentary word processor suitable for memos and letters, its real talent is as a spelling checker and on-line thesaurus. Faithful to its blue-blooded lineage, *Word Proof* works best with word processors and editors that sport the IBM logo: *PeachText*, *EasyWriter*, the *Personal* and *Professional Editors*, and Edlin. With a little prodding, however, it can handle DOS and near-DOS files created by other word processors. *WordStar* users, of course, are out of luck.

Word Proof automatically starts out in word processing mode, but its other features are just a function key away. When you finish with the spelling checker or thesaurus, you're always returned to the word processor.

The Spelling Checker. Even veterans of *Thirty Days to a More Powerful Vocabulary* will like *Word Proof's* spelling checker. With a whopping 125,000 words, its dictionary is one of the largest on the market. This means that *WordProof* doesn't waste your time by flagging a lot of correctly spelled words.

The speller gives you the option of checking a single word, a portion of your file, or an entire document. To check a single word, just place the cursor within the suspect word and press F3; to check the entire file, hit F2. Since the speller proofreads only in the forward direction, you have to start with the cursor at the beginning of your file. To check part of a file, before pressing F2, you place the cursor where you want proofing to begin.

As the program checks each word in your document, it flashes through a small scanning box on-screen. When the program finds a word it doesn't recognize, it pauses and replaces the box with a menu that gives you four choices:

- Look up possible spellings.
- Ignore the word and continue scanning.
- Consider the word correctly spelled and remember it for the rest of the session.
- Consider the word correctly spelled and quit the spelling checker.

When you choose to look up possible spellings, *Word Proof* shows you a list of words that, in most cases, includes the correct spelling. If you mistyped "learn" as "leern," for example, *Word Proof* would list "leery," "leering," "leers," "lectern," "learn," and "leper". To have "leern" replaced with "learn," you put the cursor over "learn" in the word list and press return. If it turns out that the correct word isn't the same length as your misspelled version, *Word Proof* automatically adjusts the spacing in your text. On the rare occasion when the list of possibles doesn't contain the correct spelling, you can make the correction yourself by returning to the word processor.

The spelling checker's one weakness is its handling of auxiliary dictionaries—dictionaries you create for words peculiar to your line of work. *Word Proof* lets you put no more than six hundred words in such a dictionary, a ridiculous limit considering the size of the main dictionary. One way around this limit is to create several different auxiliary word lists, one for each type of document you create (letters, briefs, and so on). However, you can use only one of these dictionaries per spell-check.

Another problem with the auxiliary dictionary surfaces when you try to add words to it. Although it's easy enough to have *Word Proof* remember a word until the end of a work session, there's no simple way to have that word saved on disk for future use. To make temporarily saved words permanent members of the auxiliary dictionary, you have to save the file you just checked (otherwise you lose the corrections), load the auxiliary dictionary into memory, and finally save the updated dictionary to disk. Most spelling checkers let you make permanent additions to the dictionary during the checking process.

Word Proof's word processor can be used to add words to the auxiliary dictionary more directly by editing the word list just as you would any file. You can also use the editor to delete those misspelled words that inevitably find their way into a dictionary.

The Thesaurus. You call the thesaurus the same way you do the spelling checker—by placing the cursor on a word and pressing one of the function keys (in this case F4). If you have single-sided drives, you have to swap your data disk for *Word Proof's* synonym disk; with double-sided drives you can get by without swapping, since you can squeeze the synonym list onto the program disk. Either way, after about ten seconds of furious disk spinning, *Word Proof* presents you with a list of synonyms. If the word can serve as more than one part of speech (as both a noun and verb, for example), *Word Proof* lists each part of speech separately. It then groups the synonyms by related meaning.

For example, if you looked up "learned," you'd see the following:

VisiCalc,® Lotus™ 1-2-3™ and Multiplan® users endorse Dow Jones Spreadsheet Link™...

"The analysis I do with Spreadsheet Link would be impossible any other way.

I track more than 200 securities. This software really gives me a handle on my investments. I can download stock quotes and fundamental information from Dow Jones News/Retrieval® directly into my spreadsheets, and analyze it using my own formulas."

Ronald Eldredge
Lockhart Financial Services

"The speed of collecting and entering data is invaluable.

Spreadsheet Link does exactly what I want it to do. It's a tremendous help in manipulating data. I think anyone who does spreadsheet analysis will find this an important addition to their software library."

Anne Freilich
Wensley Management Corporation

"The comprehensive information I get from Dow Jones News/Retrieval is a real plus.

Without Spreadsheet Link, I'd have to enter data keystroke by keystroke into my templates. This software makes spreadsheet analysis much easier and more useful to me. I'm getting very satisfactory results, and I'm still exploring the possibilities."

Douglas Gill
Grayson Gill, Inc.

DOW JONES SOFTWARE™

...*Plan on it.*

By Solutions, Inc. and Dow Jones & Co., Inc.
Compatible with Apple II and IBM PC and IBM XT personal computers.
For a free brochure, call
1-800-345-8500 ext. 165
(Alaska, Hawaii and foreign call 1-215-789-7008 ext. 165)

VisiCalc is a registered trademark of VisiCorp.
Lotus and 1-2-3 are trademarks of Lotus Development Corporation.
Multiplan is a registered trademark of Microsoft Corporation.

Data in

MicroPro.
123
LOTUS
WORDSTAR
insoft
DATA DESIGN
DATA BASE MANAGEMENT SYSTEM FOR THE IBM PERSONAL COMPUTER
insoft
Data Base Management System For the IBM Personal Computer
DATA DESIGN
DATA DESIGN
SETUP and UTILITIES
DATA DESIGN
BROWSE and PRINT

Synonyms for learned
Adjective
erudite, scholarly, scholastic, wise
Verb
get, master, pick-up
discover, ascertain, determine, find out, hear,
memorize

If there's more than one screen of synonyms ("good" has sixty-five synonyms, for instance), you use the page-up and page-down keys to view the rest.

Should you decide that one of the words in the list is just the one you're looking for, you put the cursor over that synonym and press return. The new word immediately replaces the old one in your text. If, on the other hand, none of the synonyms strikes your fancy, you can leave the original word intact and return to the word processor by hitting the escape key. When you try to look up synonyms for a word that's not in the thesaurus, *Word Proof* replies with "synonyms not available."

The thesaurus does have a few failings. Take a closer look at the example for "learned." As you can see, when you look up the regular plural or past tense of a word (in this case, "learned"), you get the synonym list for the present tense singular form ("learn"). Since you don't want any of these synonyms in your text in this tense, you have to use *Word Proof*'s editor to change the synonym into the plural or past tense once you've made the switch.

Irregular forms of a root word wreak greater havoc: *Word Proof* doesn't recognize them at all. For instance, although "run" is in the thesaurus, *Word Proof* draws a blank if you try to look up synonyms for "ran." Similarly, words such as "writing" or "easier" go unrecognized, even though "write" and "easy" are in the thesaurus. To find synonyms for such words, you first have to change them to their root form in your text. Then, after looking up the root word and substituting a synonym for it, you go back to the word processor and change the synonym to the correct form.

Word Processor. *Word Proof*'s word processor isn't going to put MicroPro out of business. Though it's fine for letters, memos, or helping your ten-year-old type her science report, try anything more complicated and you'll be begging for a few of *WordStar*'s dot commands.

One of the word processor's features is a status line at the bottom of the editing screen that displays the name of the file you're working on, the cursor's position (line and column numbers), the ASCII number of the character the cursor is on, and whether the editor is in overtype or insert mode. The insert key toggles you between these two modes.

As you'd expect, the arrow keys move the cursor a character or a line at a time and the page-up and page-down keys scroll by the screen. F8 moves the cursor to the next word; the home and end keys jump to the left and right margin and, when combined with the control key, to the top and bottom of your file. The only way you can delete is by character or to the end of the line. You can resurrect anything that you erase by accident with *Word Proof*'s unerase command.

Word Proof has very simple search and replace capabilities. It searches in the forward direction only, looking for exact matches or ignoring capitalization, whichever you choose. Replacements are by query only; there's no global replace command.

Other than setting the right margin and indicating page length, you've almost no control over your document's format. Surprisingly, however (and MicroPro, please note), *Word Proof* automatically reforms paragraphs you've edited. If you're game for creating long documents, *Word Proof* has the capability. Once you fill RAM, the program creates a spillover file on disk to handle the overflow.

Anagram Solver. *Word Proof*'s most unusual talent is finding anagrams. In case you're not a Jumble fan, anagrams are new words made by scrambling the letters of the original. For example, by rearranging the letters in star, you can come up with rats, arts, and tsar—whatever that's worth. By using the anagram feature in reverse, you can create your own word puzzles.

Documentation and Support. *Word Proof*'s documentation is thorough; the manual's one hundred ninety-six pages are equally divided between tutorial and reference sections, including a table of contents and index, a list of error messages, and a quick reference card. Patrick Henry's "Liberty or Death Speech" is provided on disk for you to practice your spell-checking (apparently Patrick didn't spell too well).

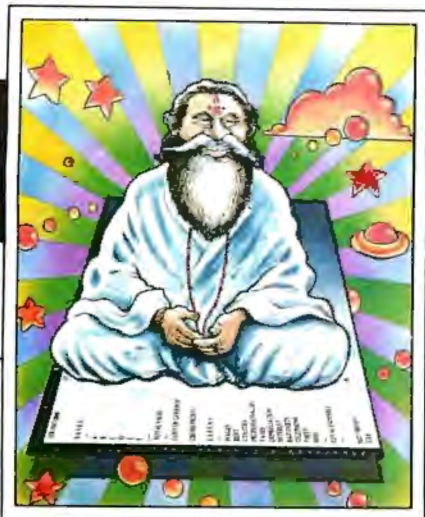
Word Proof isn't copy-protected.

Audience. Anyone who knows a bargain.

System Requirements. *Word Proof* requires at least 96K. With one disk drive you can use everything but the thesaurus; for that, you need two drives. *Word Proof* runs on PC*jr*. ▲

Wordvision
Bruce & James Program Publishers
The Wharfside Building
680 Beach Street, Suite 357
San Francisco, CA 94109
(415) 775-8400 Price: $79.95
Word Proof (version 1.0)
IBM
Box 1328-W
Boca Raton, FL 33432
(305) 998-2000 Price: $60

The Spreadsheet Guru



by Jack Grushcow

What Makes *1-2-3* So Good?

Until now, all the models presented in this column have been simple and general enough that you could create them with any spreadsheet program—*VisiCalc, SuperCalc, Multiplan*, or whatever. Over the course of the next several installments we will be focusing instead on *1-2-3* and we'll be addressing ourselves to some of the unique capabilities provided in that program.

Before we get down to specifics about working in *1-2-3*, however, let's look at this program—and its publisher—in a general way. As you read this discussion, you may notice that it says almost nothing negative about *1-2-3*. Please bear in mind that the author is in no way related to Mitch Kapor or affiliated with Lotus Development Corporation.

Now on to the facts.

The Company

The success of *1-2-3* is attributable as much to marketing savvy as it is to beautiful software engineering. After all, in these days of vicious competition for the software buck, it's not enough to build a better mousetrap. You have to sell that trap, too. Lotus has demonstrated expertise in both areas.

A consideration of the company behind a software product is crucial to a full evaluation of the product itself. This is especially true if the person doing the evaluating is also in a position to influence others' software choices.

Learning any piece of software requires an investment—sometimes a major one—of time and effort. When you decide to adopt a software package, you are investing in the software as well as the software supplier. It's reasonable to expect updates, fixes to bugs (if any), ongoing support, and access to a growing product family. A company that has the staying power to be in business for the long run is in the best position to offer you these values.

Judging by the best information available today, it's safe to bet on Lotus's being around a while.

Another important consideration is a software company's philosophy, or style. You'll notice that successful software publishers tend to develop an individuality in the way they address the user and the computer—a distinctive way of doing things that runs consistently through their entire product line. The way a software package approaches a problem can affect your life, because you often end up thinking in its terms, which may or may not be your own. If, for example, a program offers to organize your desk, you can be sure it will necessitate some adjustments to your standard mode of operation. If a program records your daily schedule, certain of its features may have you booking certain types of appointments in the morning rather than in the afternoon.

The moral is clear: If you think you'll be working with more than one product from a particular company, be sure you're comfortable with that company's style—you may have to live a few years with it.

If you're not convinced that the way a product behaves should be a key factor in your decision making regarding software, you've probably never had the chance to attend a software product planning group. Here the order of tasks and the style of screen presentation can claim the level of attention that Givenchy lavishes on a model before a crucial fashion show. And such issues can divide members of a software planning group with the fervor of religious or political conviction.

The new windowing products present some good examples of the kind of questions software designers argue over. Is it better to start the program with a single large empty window or with several empty windows? Should the windows all initially display data? When data is to be moved from one window to another, is it better to have the user first select the window that will receive the data or should the user first define the data he wishes to send?

The sum of the answers to all such questions is a software product.

All of this discnssion is preamble to the following, admittedly subjective but strongly felt, assertion: *1-2-3* has a nice style. It's *gemütlich*, comfortable, effective, and easy to work with. It thinks the way you do and works with you to solve problems.

Let's illustrate the point. We'll compare the way labels are entered in *1-2-3* with the way they're entered in one of *1-2-3*'s close competitors, *Multiplan*.

Suppose your cursor is in cell A9 and you want to move it to A12 and enter the label LEGAL FEES. Using *Multiplan*, you would type:

[Enter] LEGAL FEES [Enter]

After your cursor is over A12, you have to press the enter key to select the *Alpha* option from the *Multiplan* menu. Only then can you enter a label. If you forget to press enter before you type the words *LEGAL FEES*, *Multiplan* will think you're making a series of menu selections L, E, G, and so on. Not only will you not get a label, you'll get beeped at to boot!

Lotus recognizes the fact that spreadsheet users type lots of labels. To enter the same label in *1-2-3*, you type:

LEGAL FEES [Enter]

No push of the enter key is required before label entry.

Now this may seem like a trivial point, but it isn't (and it's by no means the only way in which *Multiplan* is less streamlined than *1-2-3*). It's amazing how quickly one becomes a demanding, dictatorial user when it comes to software—even if one starts out as a nervous neophyte. And once you've become proficient at using a product, you're likely to find such things as extra keystroke requirements as irritating as discovering that your new hibachi kit is missing two screws.

When it comes to software design, a single keystroke can make the difference between fortune and failure.

Ease of Use

Lotus makes your life easy in lots of ways, and both the novice and the advanced user benefit from the product's simplicity. Let's look first at what *1-2-3* offers the novice.

Most computer trade shows are like a 3 ring circus...

PCExpo is the only trade show for serious IBM PC-compatible buyers.

Large volume corporate buyers
OEM's
Systems houses
ISO's
Value adders
Resellers
Software developers
Other large volume buyers

At those oversized computer trade shows everybody and his brother is hawking his wares. So you waste half your time wading through irrelevant displays and meaningless hype. PCExpo is the most sharply focused of all computer trade shows...only PC's and PC compatibles...and not a dealer or distributor booth in sight! You'll find factory people and factory literature...hardware, software publishers and principal service vendors. You'll get the concerned attention of hundreds of designers and manufacturers of thousands of IBM PC compatible products. Make a hands-on survey of the industry at PCExpo and you'll make an informed decision on your PC purchases. So plan to be there!

1200 booths on 3 big floors at the NEW YORK Coliseum, June 26–28, 1984

PCExpo

333 SYLVAN AVENUE, ENGLEWOOD CLIFFS, NJ 07632

IBM is a registered trademark of International Business Machines Corporation

SAVE TIME AND MONEY—USE THIS COUPON!

ADVANCE REGISTRATION FOR PCExpo • NEW YORK COLISEUM, JUNE 26-28, 1984

Use one form per person. Photocopy if necessary.

NAME

BUSINESS TITLE

COMPANY

COMPANY ADDRESS

CITY STATE ZIP

MAIL STOP (if any)

TELEPHONE

2

PCExpo attendees must be qualified. Please check the applicable boxes in two of the appropriate fields below. (Minors may not register.)

TRADE RESELLERS - YOUR COMPANY'S MAIN BUSINESS ACTIVITY

I
- 1 ☐ Computer consultant
- 2 ☐ Computer dealer/dist.
- 3 ☐ Computer OEM
- 4 ☐ Office products dealer
- 5 ☐ Service vendor
- 6 ☐ Software developer
- 7 ☐ Systems house
- 8 ☐ Turnkey vendor
- 9 ☐ Value adder
- 10 ☐ Other (specify)________

CORPORATE VOLUME BUYERS - YOUR COMPANY'S MAIN BUSINESS ACTIVITY

II
- A ☐ Accounting firm
- B ☐ Advertising
- C ☐ Banking
- D ☐ Communications
- E ☐ Construction/Architects
- F ☐ Credit
- G ☐ Education
- H ☐ Engineering
- I ☐ Government/Military
- J ☐ Hospital
- K ☐ Hotel
- L ☐ Industrial design
- M ☐ Insurance
- N ☐ Law office
- O ☐ Management consultant
- P ☐ Manufacturing
- Q ☐ Publishing
- R ☐ Real estate
- S ☐ Research/ Development
- T ☐ Retail sales
- U ☐ Securities broker
- V ☐ Transportation (all)
- W ☐ Utility
- X ☐ Wholesale/Retail sales
- Y ☐ Other (specify)________

YOUR JOB FUNCTION (Check main one only)

III
- AA ☐ Accountant
- BB ☐ Administrator
- CC ☐ Consultant
- DD ☐ Corporate officer
- EE ☐ Creative arts (all)
- FF ☐ Designer (all)
- GG ☐ DP/WP manager/ operator
- HH ☐ Engineer (all)
- II ☐ MIS Dir./Mgr.
- JJ ☐ Programmer
- KK ☐ Purchasing
- LL ☐ Sales/marketing
- MM ☐ Scientist
- NN ☐ Securities analyst
- OO ☐ Service technician
- PP ☐ Systems Analyst
- QQ ☐ Other (specify)________

ADVANCE REGISTRATION FEES - EXHIBITS & MEETINGS

1 day - $20. Date must be given here ________

2-3 days - $40.

Mail check payable to PC Expo with completed form to:
PCExpo 333 Sylvan Avenue
Englewood Cliffs, NJ 07632
Tel: (201) 569-8542

Your show credentials will be mailed on or before June 8, 1984. Advance orders must be received by PCExpo no later than June 4 for return mail service. Advance orders after that date will be held at the "Will Call" window for your arrival. No foreign advance orders. All registrations must be accompanied by full payment, and are non-refundable.

On-site registration fees are $25 for one day, $50 for 2-3 days. No checks accepted on site.

The Tutorial. *1-2-3* comes with the most interactive, simple-to-use, well-written on-line tutorial ever produced for a microcomputer product. It's divided into several sections and takes someone who has never before used a spreadsheet from the fundamentals of cursor movement all the way to advanced data management and graphing. It doesn't lose anyone along the way, either.

The tutorial gives you a nonthreatening environment in which to learn your way around the spreadsheet. Its presence means that you don't have to crack the large, possibly intimidating manual until you've had time to develop some familiarity with the program. You can't appreciate just how effective this feature is until you see an executive who has never used a computer, much less an electronic spreadsheet, move through the *1-2-3* tutorial, gaining confidence with each new screen.

The Menus. It's easy to forget the way Lotus menus work after you have used the product. A good user interface, like a good pair of skis, dissolves into the background even as it does its job.

You can select all Lotus menu options simply by using arrow keys. (*Multiplan*, by contrast, presents the user with a confusing array of keys for moving to and selecting menu options.) Lotus menus are not only easy to use, they're informative, too. When you move the cursor to highlight a command option, you get a brief explanation on-screen of what the command does. This is the type of feature that's there when you want it but doesn't get in the way once you know what you're doing.

Not only are the commands easy to locate, but the order in which they're arranged within the various menus makes sense as well. Options that you're most likely to select appear ahead of options that you'll use less frequently.

The Help Function. No less impressive than the tutorial is the built-in help function. From any point within *1-2-3* you can get information about the command you're working with by pressing the F1 key. The spreadsheet will disappear, and your screen will display a description of the command you were working with at the time you hit F1.

1-2-3's help screens appear quickly and contain highlighted keywords. Moving the cursor over these keywords and pressing the enter key gives you access to additional help screens about related topics. When you're done getting help and want to return to work, all you have to do is press the escape key and you're back in the spreadsheet—right where you were before you hit F1. The help feature is so fast and effective that you won't have to reach for the user manual if you ever get stuck in the program.

Integration. Integration means the ability to send data from one application to another in a quick, painless way. Take, for example, the manager who starts working out a budget on a spreadsheet. After several tries at the budget, inserting new numbers and seeing what effect they have on the overall plan, the manager settles on a specific scenario.

Now he or she may want to draw a graph using data in this spreadsheet. After all, there's no better way to get a feeling for a forecast than to see how it looks graphically. Often this kind of examination leads to further adjustments in the spreadsheet; data gets changed, the graph gets redrawn, and the result is what's called a decision-feedback loop.

Now imagine that after the budget has been adopted the manager wants to prepare a letter to his subordinates. He may want to tell them about the great year they are all about to experience, or he may wish to exhort them to hang tough and maintain tight control over spending. Either way, it's natural to assume that he would like to include some or all of the budget data in his letter. Probably the graph, showing certain key sections of the budget information, will help the manager make his point.

Without going further into this example, we can see that the need to exchange data between a spreadsheet, a graph program, and a word processor is a common one.

Until recently, the only way to transfer data between spreadsheets and other programs was through a type of file, called the Data Interchange Format file (DIF), created by Software Arts and first used in *VisiCalc*. While it looked useful in theory, the DIF format was about as practical as a pair of papyrus jogging shoes.

The goal of the newer integrated software packages is to allow you to move quickly and easily between different types of applications. Lotus takes some initial steps toward this goal by supplying you with a super spreadsheet that has graphing and some rudimentary word processing and data management built in.

Let's examine some of the data management features. The database functions in *1-2-3* can be divided into two groups: the database commands and the database statistical functions.

The database commands are instructions that allow you to manipulate data in a predefined section of the worksheet. *1-2-3* lets you designate blocks of data in your worksheet to be a database. In a series of adjacent cells, each individual row becomes a database record, and each cell within that row becomes a field.

For example, a simple *1-2-3* address book might contain a list of names, addresses, and

phone numbers. Each row would contain one person's name, address, and number. So each line of data is a record, and each individual name, address, and phone number is a field within that record.

Using the *1-2-3* database commands, you could ask for a search to be performed on a name, and the record that contained that name (as well as the corresponding address and phone number fields) would be highlighted. There's another, better, way to get this information; it involves use of the Data Query command and will be discussed in a subsequent installment of this column.

The second type of database functions, the database statistical functions, are a logical extension of the familiar @SUM and @MIN spreadsheet functions. The statistical function @DSUM, for example, lets you total only that column in a database that corresponds to a criterion you've selected. Let's see how this works.

Look at the following spreadsheet:

	A	B	C
1	Name	Sales	Division
2	Jones	120	1
3	Smith	300	2
4	Adams	250	3
5	Green	340	2
6	Tropp	255	1

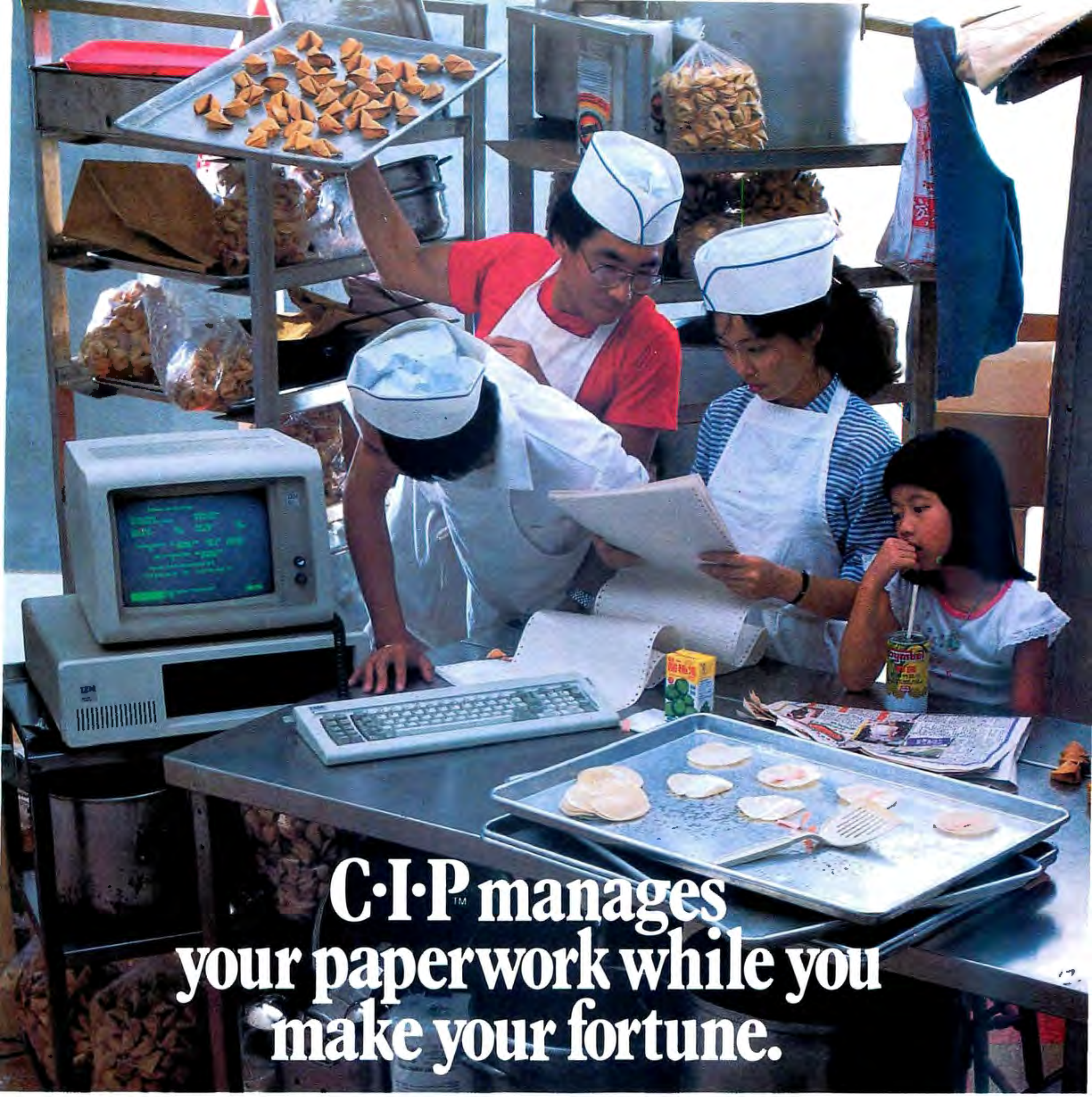
The block of data from A1 to C6 can be defined as a database. Once this has been done, the database functions can be used to select and display database information.

Each of *1-2-3*'s database functions requires three arguments: input range, offset, and criterion range. The input range specifies the block of data to be used as a database. The offset range defines which column of the database is to be used by the function (the first column in the input range is called column 0). The criterion range is used to describe those records that are to be acted upon by the function.

In our little example here, we can use the @DSUM function to find sales totals for each division. Let's say we wanted to find the total sales for division 1. Our input range would be A1. . .C6. Our offset would be 1. And our criterion would be that the division must equal 1.

In subsequent articles we'll explore the use of each specific database command in *1-2-3*. For now, we simply want to describe in a general way the types of functions available.

The key limitations of *1-2-3*'s database capability that you should be aware of are that records are limited in length to thirty-two fields, that all files are flat (no hierarchies are possible), and that when the data block is large, *1-2-3* can be quite slow in carrying out these functions. ▲

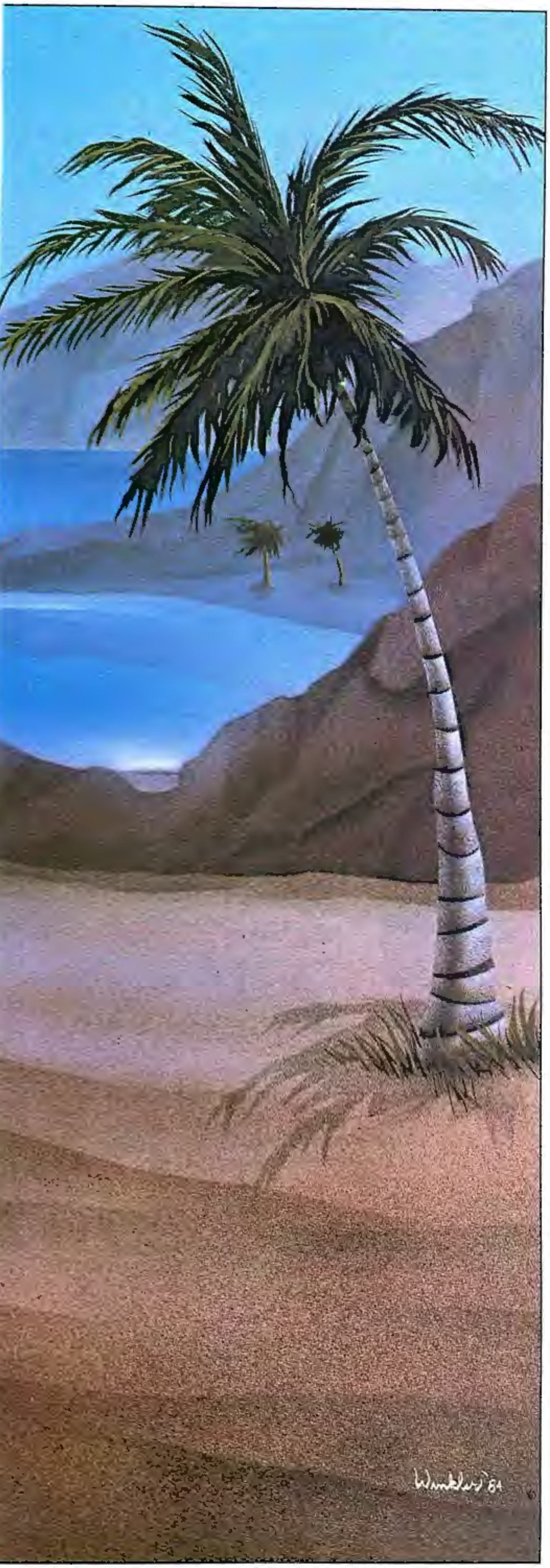


In businesses of almost every size these days, the more successful you are, the less time you have to spend with paperwork. Now you can turn all that paperwork over to C·I·P™, the Concentric Information Processor, and your IBM® PC. • You'll be using C·I·P confidently and profitably within hours. Designed for first-time users, C·I·P lets you create files, alter formats, and design reports visually. So simple! Yet C·I·P has the flexibility to adapt to changes in your business and the power to grow along with your needs. • Ask your local IBM PC dealer to show you how C·I·P can keep inventory under control, manage mailing lists, write reports, calculate due dates, figure sales tax – in other words, take over the paperwork, while *you* concentrate on running the business and increasing profits. C·I·P from Concentric Data Systems, 18 Lyman Street, Westboro, MA 01581, (617) 366-1122.

C·I·P™
CONCENTRIC INFORMATION PROCESSOR

The new standard for information management software.

C·I·P is a trademark of Concentric Data Systems, Inc.
IBM is a registered trademark of International Business Machines.



A Review of Four Systems

Logos for the IBM

by Mark Bridger

No, International Business Machines is not looking for a new symbol for its personal computer: The Logo in question here is a programming language, and this article is a review of the four versions of Logo currently available for the PC. The appearance of Logo for the PC is especially noteworthy since, with the exception of mainframe versions, the only Logo language available has been for eight-bit machines. The larger memory of sixteen-bit machines such as the IBM has led to a number of enhancements of the language as well as to increased room for Logo programs.

What Is Logo?

Historically, Logo is a direct descendent of Lisp, the principal programming language of artificial intelligence research. So, to provide some background, we'll start this review with an overview of Lisp.

In Lisp the fundamental objects to be studied and manipulated are *lists* (hence the name Lisp, which is a contraction of *List Processing*). A list is an ordered collection of objects, which may be identifiers (such as Toad, or Position, or X), or numbers (such as 2 or 128.345), or other lists. A list is usually enclosed in parentheses or brackets. Here are some examples:

[TOAD, POSITION]	(two-element list)
[]	(empty or zero-element list)
[TOAD, POSITION, 2]	(three-element list)
[TOAD, [POSITION, 2]]	(two-element list whose second element is itself a list)

Lisp and Logo share many elementary list-handling capabilities. To demonstrate these shared operations we will use the notation from Logo.

You may create lists by piecing together discrete objects, using the *list* command. So, for example,

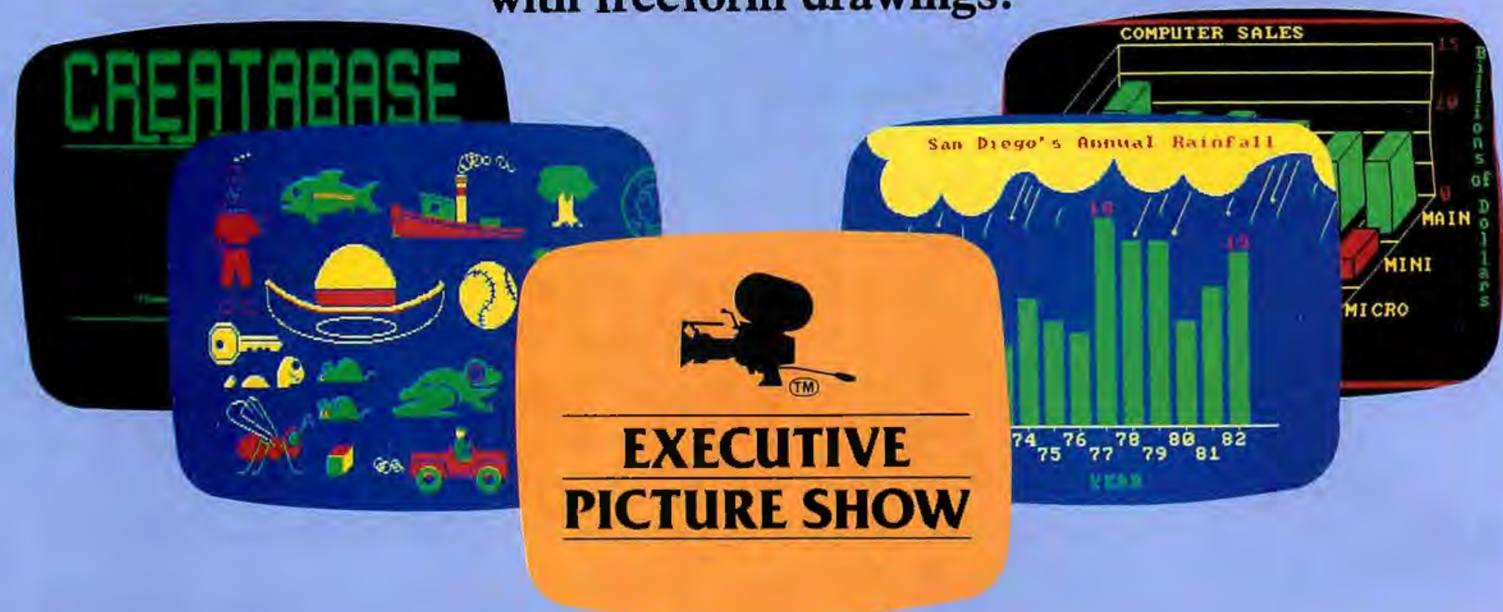
(LIST "MAPLE "OAK "HAWTHORN "CHERRY)

produces the list:

[MAPLE OAK HAWTHORN CHERRY]

(where the quotes are a requirement of Logo syntax). You can also produce lists by concatenating elements of other lists, via the *sentence* operator.

Introducing a graphics system to meet your every need, whether it's creating sophisticated presentations that integrate screens from your favorite programs — or expressing yourself with freeform drawings.



What Lotus™ 1-2-3 has done for spreadsheets, Executive Picture Show™ is doing for graphics.

Ordinary business graphics programs are fine as far as they go. But they are hardly the stuff of interesting business presentations.

If you need the ability to integrate and modify screens from other programs, such as Lotus 1-2-3, dBASE II®, CREATABASE, and WordStar™ — or just give free rein to your artistic side with free-form drawings — you've probably been frustated on both counts.

Now there's a business graphics system that gives you these capabilities, plus the usual line, bar, 3D bar, horizontal bar, and surface and pie charts.
It's the Executive Picture Show and it's long on capabilities where other programs fall short.

With Executive Picture Show you can create:

- free-form graphics
- business graphics
- slide show presentations
- animated presentations

Presentations to keep viewers on the edge of their seats.

Not only does Executive Picture Show accept screens from other programs, it lets you integrate them into your presentation where and when you want them. Then you can add the sound, motion, and color that insures a captive audience. Dropping in your company logo or making bar graphs take form right before your viewer's eyes is easy as pressing a few keys.

Executive Picture Show allows you to use both your IBM® monochrome and color monitors during your presentation. This means you can show a spreadsheet on your monochrome monitor, while a graph or drawing is formed on your color display.

Interactive presentations.

The Executive Picture Show was designed with you and the audience in mind. Not only does it allow you to design a moving presentation, it gives your viewers a chance to respond with more than enthusiastic reviews. They can actually input their responses so the program — and you — can act on their input.

Easy to use.

We had the business executive in mind when we designed this tool. That's why Executive Picture Show cuts through all the "computer-ese" to simplify instruction and prompts you throughout the program. We've reduced many steps to single key commands and included a handy reference card to help you get your show on the road.

Executive Picture Show is playing at a computer store near you for jnst $195. If you want to preview this program, contact PCsoftware of San Diego directly for a demonstration disk and documentation priced at $30.

Requires: Graphics adapter and display
128K RAM
2 disk drives or hard disk
IBM PC or IBM XT

Lotus 1-2-3, CREATABASE, and WordStar are trademarks of Lotus, PCsoftware of San Diego, and Micropro International Corp., respectively. dBASE II and IBM are registered trademarks of Ashton-Tate and International Business Machines, Inc.

Dealer inquiries invited.
VISA and Mastercard accepted.

Dealer orders contact:

Micro D	(CA)	800-432-3129
	(Nat.)	800-854-6801
Software Distributors	(CA)	800-252-4025
	(Nat.)	800-421-0814
Vitek	(CA)	800-237-7290
	(Nat.)	800-237-3443

Also available through Computerland Corporate.

PCsoftware™

PCsoftware of San Diego
Suite 416
9120 Gramercy Drive
San Diego, CA 92123
(619) 571-0981

(SENTENCE [TOAD POSITION] [MAPLE OAK HAWTHORN CHERRY])

produces the list:

[TOAD POSITION MAPLE OAK HAWTHORN CHERRY]

The most basic list operations are *first, butfirst, last, butlast, fput,* and *lput*.

First picks out the first element of a list, and *last* selects the last element:

FIRST [TOAD, POSITION]

outputs

TOAD

and

LAST [TOAD, POSITION]

outputs

POSITION

Butfirst and *butlast* output lists with all but the first or all but the last element of the original:

BUTFIRST [TOAD, POSITION, 2]

outputs

[POSITION, 2]

and

BUTLAST [TOAD, POSITION, 2]

outputs

[TOAD, POSITION]

Fput and *lput* require two inputs: an object and a list; they output new lists formed by placing the object at, respectively, the first and last positions in the list:

FPUT "TOAD [POSITION, 2]

outputs

[TOAD, POSITION, 2]

and

LPUT "TOAD [POSITION, 2]

outputs

[POSITION, 2, TOAD]

(The " before the identifier Toad is a syntax requirement in Logo.)

Clearly, with combinations of these operations we can create or take apart any list in any way we want.

One really striking and important feature of Lisp (and Logo) is that a program can be considered as a *list* of instructions, and as such can itself be manipulated by other programs or by itself.

Lest you get the wrong idea, we should immediately point out that there's much more to Lisp than just these simple list-manipulating operators. Lisp "operates" by allowing the programmer to create, from primitive operations such as these, functions and procedures that can be strung together to create more complicated functions and procedures. These functions and procedures, moreover, can call on themselves, even in the statements that define them (see the Hilbert Curve program appended to this review). Furthermore, contemporary versions of Lisp include other data structures, such as arrays, in addition to lists.

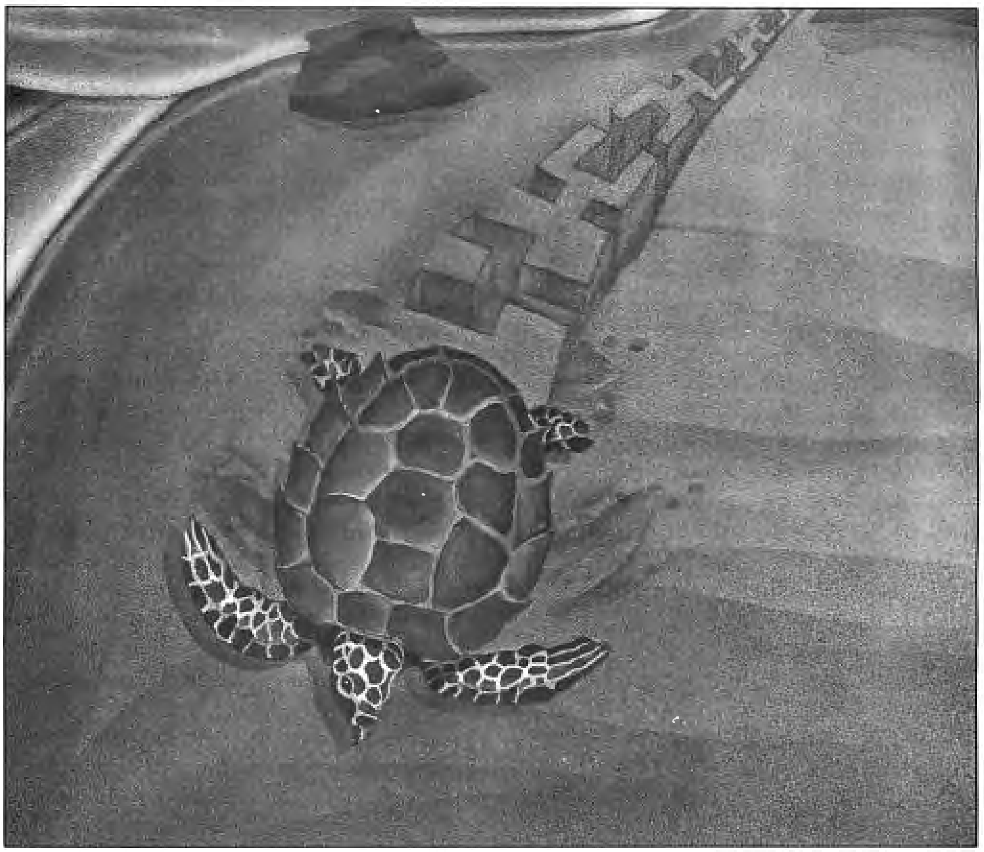
For more information on Lisp, take a look at *LISP* (Winston and Horn, Addison-Wesley, 1981), a standard and quite readable text.

Turtle Graphics

Lisp was invented by J. McCarthy in the early 1960s and has seen many refinements and enhancements since then. One of its offshoots was Logo, which was initially a "dialect" of Lisp. However, the real revolution came with the development of the "turtle." This was initially a small tortoise-shaped machine of Plexiglas and metal complete with wheels, a pen, and an umbilical cable. It responded to typed commands such as *forward 20, right 15* (degrees), or *backwards 40* by moving in the appropriate direction. *Pendown* caused it to lower its pen, resulting in a visible trail of its progress along the floor. The list of commands

[PENDOWN FORWARD 20 RIGHT 90 FORWARD 20 RIGHT 90 FORWARD 20 RIGHT 90 FORWARD 20 RIGHT 90]

for example, would cause the turtle to draw a square on the floor and end up in its original position and direction.



The remarkable thing about the turtle was that programming it was so fascinating and challenging that people who tried it forgot whether they feared or disliked math and/or computers. This was especially the case with children. Professor Seymour Papert, a noted educator and for many years head of the MIT Logo Research Group, has written a book called *Mindstorms* (Basic Books, 1980) in which he describes quite elegantly the important educational potential of Logo.

Soon the turtle was available on the cathode-ray tube as well as the floor and "Turtle Graphics" became a standard part of Logo; in fact, Logo has generally become so associated with Turtle Graphics that its list-processing capabilities tend to be (unfairly) ignored.

Apart from its Lisp ancestry, Logo bears a similarity to Basic in that both are interpreted languages (as is Lisp). This means that execution of commands is rather slow, since each command has to be translated into machine language every time it's encountered in a program.

Logo also shares some important features with Pascal: Both languages allow subprograms called "procedures"; both allow "recursion" (that is, they allow a procedure to call itself); and both are highly "structured." A structured language is characterized by the absence of line numbers; no line numbers are needed, because a program's syntax (structure) tells it what to do when. A good example of these features is provided by the Hilbert Curves program listed at the end of this article.

Finally, because of the way Logo is implemented on computers (using data structures called *nodes*) and because the whole Logo interpreter must "reside" in memory while Logo is operating, Logo requires a *lot* of memory if you want it to run anything but the most trivial programs. 64K seems like a minimal amount, and 128K or more is advisable.

The Logos Available

The four Logo languages available at this time for the IBM PC (and/or PC*jr*) are:

DR Logo (Digital Research): $150; requires 192K and can use up to 256K.

IBM Logo (IBM/Logo Computer Systems): $175; requires 128K and can use at least 192K.

PC Logo (Harvard Associates): $150; can use 64K with some features sacrificed, but full implementation requires 128K.

Waterloo Logo (Waterloo Microsystems): $125; requires 128K and can use 192K.

The rest of this article will compare these four software packages by grouping their features within several categories; afterward we'll try to draw some general conclusions.

Instructions and Documentation

All four Logos have well-written descriptions of their primitive procedures. IBM, PC Logo, and DR Logo have fat looseleaf books containing user guides, reference manuals, cutesy turtle hints, and numerous sample programs; DR Logo even has color plates illustrating the output of graphics procedures. Waterloo Logo supplies the more modest, spiral-bound "Tutorial and Reference Manual," which, while not as extensive as the other manuals, is adequate and at least does not crowd the computer off your desk; a more extensive revision is promised soon. IBM provides the separate, spiral-bound "Logo Programming with Turtle Graphics"—a fairly complete introductory guide to using the turtle to draw pictures and patterns.

Each of the books has a reasonably good index; Waterloo and IBM include separate, small fold-up reference cards listing all of their primitive procedures and functions. The PC Logo book is a little hard to use, since the procedures are grouped by type and only arranged alphabetically within types. With the possible exception of Waterloo Logo, any of the four has as good (or better) documentation as IBM provides with its Basic.

Editing and Debugging

Here we have a mixed bag. All four packages have workable, full-screen editors. The Waterloo and PC Logos have superior enhancements: Waterloo provides a nice procedure for delineating and moving blocks of text (using reverse video), while PC Logo has a less impressive block move (using the erase buffer) but provides a search-and-replace facility that can be very useful for debugging and changing procedures, especially in long programs.

By contrast, DR's editor is spartan, providing just the usual cursor moves and page-up/page-down commands. Since DR Logo is not DOS-compatible (more on this later), it probably can't read files prepared with your text editor or word processor; that deficiency makes its lack of advanced editing features even more significant.

The editor provided by IBM is also without special features, but since IBM Logo is DOS-based, any editor can be used to prepare extensive programs. Use of an outside editor is inconvenient, however, since it means leaving Logo, going to the editor, saving the program, returning to Logo, and loading the program again—all before you can test your program.

DR, however, recoups when it comes to debugging. DR Logo provides a nice set of interactive debugging commands that enable you to go through a program step by step, noting which procedures are called ("Watch"), checking the values of all variables ("Trace"), and even splitting the screen between these values and the output produced by the procedure ("Debug"). Furthermore, this facility also prints out the "level" of each procedure call— whether it's a "top-level" procedure, a subprocedure of a procedure, or a subprocedure of a subprocedure of a procedure, et cetera.

The most recent version of PC Logo provides a Trace command to allow single-stepping. The other two Logos provide only a Pause command, which stops the program, allowing one to inspect the variables (print them out) at a given point in program execution.

Graphics

Here again, a mixed bag of properties, which we summarize below.

Wrap allows the turtle to leave one side of the screen and appear on the opposite side; *wrap* can be turned on or off. This feature is provided by IBM, DR, and PC Logos.

Fence prevents the turtle from leaving the visible screen; it's the only turtle mode for Waterloo, and it's an option on the other systems. Also optional on the others is *Window*, which lets the turtle move even when outside the screen area and hence the turtle's invisible.

Arc/circle drawing allows specification of an arc by its radius and a number of degrees, or a circle by its radius and center. These are primitive (built-in) commands in Waterloo Logo; IBM's manual provides written programs for creating them.

A *screen dump* is a procedure allowing you to print the picture currently on the screen on a dot-matrix printer (in this case an IBM or Epson). All four Logos provide this.

Fill is a command that fills an enclosed region containing the turtle with a solid color (it's equivalent to *paint* in IBM or GW Basic). This is provided by Waterloo and IBM (and promised by Harvard Associates for a later version of PC Logo). A forthcoming enhancement of Waterloo Logo will contain a *fill* command that will paint in various "textures" as well as various colors.

*High-res*olution graphics provide a 640-by-200-dot screen in one color over black. Only PC Logo offers this (and it's not available for PC*jr*); the others (including PC Logo) use 320-by-200 four-color screens.

Finally, while not strictly in the graphics category, all four Logos provide a primitive command that produces a *tone* of a specified frequency and duration. This, together with instructions in the manuals, provides music-making capabilities.

List Processing

The Logos under consideration all offer the list-handling primitives described above as well as some enhancements. One enhancement is the ability to attach to the elements of a list a property ("color," for example) and to specify for each element a value or set of values for this property ("blue," for example). Having exercised this capability, one can pick out certain elements of a list on the basis of their possessing or not possessing a certain property value.

While all four Logos have fully implemented property capabilities, DR Logo seems to have the inside track on other list-handling features, providing such niceties as *shuffle*— creating a random permutation of the elements of a list; *sort*—sorting a list in ascending order; and *piece*—picking out a segment of elements (say, third through ninth) of a list. DR also has a very handy command that converts its word argument to all uppercase or all lowercase. While one can program these functions in the other Logos, it's nice to have them as primitives. The DR Logo manual has a very amusing random recipe creator that makes use of the *shuffle* routine. It's no problem, however, to write a program to convert English into pig Latin in any of the Logos, using procedures provided (this seems to be a popular sample program for manual writers).

Arithmetic

Although Logo is not generally considered a language in which to do arithmetic, there's no reason it can't be as powerful a number cruncher as Basic (although its stack-oriented recursion can slow it down some). Unfortunately the designers of Logo often omit some of the standard functions needed for scientific purposes; for example, neither PC Logo

123™

WE KNOW HOW TO COUNT THAT'S WHY...
WE MADE A 5 in 1 PROGRAM!

DOS 1.1 or 2.0 CP/M Z-80 Version Also FOR THE:

THE ULTIMATE

1 WORD PROCESSOR
- Menu Driven
- Complete Cursor Movement
- Footnoting
- Page Numbering
- Global Replace and Search
- Many Other Powerful Features

2 DICTIONARY
- 13,000 Words
- Add Words
- Delete Words

3 MAIL MERGER
- Personalized Letters
- Lists
- Labels

4 Z-COM
The Power of Electronic Mail is Yours Now with Your Modem and **THE ULTIMATE.**

You can send letters, invoices and memos anywhere in the country within 48 hours without licking a stamp.

5 DATA BASE
- On Screen Formating
- Reports
- Form Fill In
- Column Movement
- Sorting

Computer Creations

WHY BUY EACH PROGRAM SEPARATELY? YOU CAN HAVE IT ALL FOR ONLY ~~\$385.00~~ (suggested retail)

Now \$249.50

Some of the machines "THE ULTIMATE" runs on:

CORONA-P/C
COMPAC-P/C
COMPAC-PLUS
COLUMBIA-P/C
CHAMELEON-P/C
PRONTO-P/C
DIMENSION-P/C
TEL-VIDEO-1605
LANIER-LBP 1000
OLYMPIA
PANASONIC-SENIOR PARTNER
WORDPLEX-PORTABLE
WORDPLEX-DESK TOP
IVY MICROCOMPUTER-3001
IVY MICROCOMPUTER-3002
IBM-P/C
IBM-XT
IBM-PC/JR
EAGLE-P/C
EAGLE-SPIRIT
TAVA-P/C
HYPERION-P/C
SUPERBRAIN
COMPU-STAR
KAY-PROII
KAY-PRO IV
TOSHIBA-T 300
DATA GENERAL-MODEL 10 (DESK TOP GENERATION)
HERO
PWS (PERSONAL WORK STATION BY LEE DATA)
AVATAR

New ones are being added regularly. Check with your dealer.

IF WE MADE IT ANYMORE...USER FRIENDLY IT WOULD BE ILLEGAL

COMPUTER CREATIONS INC. 766 El Camino, San Carlos, CA 94070 • 415/595-4466

nor Waterloo has a logarithm or exponential function, although all four have sine, cosine, and arctan. This is particularly surprising, since PC Logo has built-in 8087 support and is therefore particularly suited for floating-point functions.

Furthermore, Waterloo and DR Logo both offer double-length integers, while PC Logo stops short at integers of −32,767 through 32,768. On the other hand, PC Logo offers integers to arbitrary bases as well as logical *and/or/xor* (that is, bit-level manipulation) of integers. It also offers logical shift left, which is useful for fast multiplication by powers of 2.

All four Logos offer the usual *infix* operators (+, *, /, −), as well as the prefixes *sum*, *product*, *difference*, and *quotient*; Waterloo, though, has the annoying requirement that negation of numbers be written with a tilde instead of a hyphen.

Operating System and I/O

This category points out sharp differences in the four Logos. The PC and IBM Logos are absolutely MS-DOS-oriented; in fact, PC Logo has procedures enabling the user to call directly all DOS interrupt 21H functions, with direct access to the 8088 registers, while IBM Logo provides a way of loading and calling machine language subroutines that can do the same. Furthermore both PC Logo and IBM give access to DOS commands (*copy*, *dir*, and so on) as well as allowing redirection of the I/O stream. IBM also provides the *.examine* and *.deposit* commands, which are the Logo versions of Basic's *peek* and *poke*.

At the other extreme is DR Logo, which has nothing at all to do with DOS. Since DR Logo operates under CP/M, that operating system must provide your I/O and disk management. No DOS compatibility exists at all in DR Logo, and DR Logo cannot use any DOS files. In itself this is no tremendous disadvantage, since Digital Research has provided enough of CP/M to take care of all of Logo's needs. However, it does mean that if you use DR Logo you can't copy someone else's Logo program, saved on disk, unless it too was prepared on DR Logo. Furthermore, given the growing ascendancy of DOS as a microcomputer operating system and the fact, pointed out above, that Logo users may prefer their own text editors for writing long programs, DR's refusal to read DOS files may turn out to be a real disadvantage.

In the middle is Waterloo Microsystems, which also has its own operating system. Since this system, PORT, isn't yet on the tip of everyone's tongue, the company decided, quite wisely, to make DOS files accessible to its Logo. Although Waterloo doesn't, at this time, offer the plethora of commands available to users of PC and IBM Logos, it does provide some nice touches—such as a choice between scrolling text or viewing it screen by screen. An expanded version of Waterloo Logo is said to be imminent.

PC Logo and DR Logo both support light pens, while DR Logo and IBM support joysticks.

Performance and Extras

In the category of speed, PC Logo may be the clear winner, especially as far as graphics and real-number manipulation are concerned. The reason for this is that PC Logo has built-in support for the 8087 NDP chip. This is the chip that can be seated next to the 8088 on the IBM's motherboard and that provides lightning-fast floating-point calculations. You may wonder why this is important for a language like Logo whose operation seems to be centered on graphics. But that is just the point: Logo, like Basic, assumes its variables to be real. Points to be plotted are assumed to have real coordinates, and the computations required to move the turtle on a straight line are floating-point calculations.

To determine what difference the 8087 might make, I typed in a program to draw rather well-known mathematical designs called Hilbert Curves (the program appears at the end of this article). These involve recursive procedure calls and draw only horizontal and vertical line segments (see diagram). The timing results were quite dramatic: What took fifteen minutes in DR Logo, nearly twelve minutes in Waterloo Logo, and five minutes in IBM Logo took only one minute in PC Logo! On a machine without the 8087, these differences vanish; so for users who don't have an 8087 and don't plan to get one, PC Logo loses this advantage.

On the other hand, it is by no means clear that speed is always of major importance in Logo. Sometimes, especially with children's programs, it can be instructive to see *how* the turtle draws a programmed picture: The sequence of moves can be illuminating. If the picture is drawn too quickly, one may not appreciate what has happened or have time to enjoy the act of creation. Therefore, it's advisable to take timing benchmarks with at least a grain of salt.

In the category of extras, PC Logo provides a nice utility disk that includes the famous animal guessing game (in which the computer tries to guess an animal you're thinking of and learns from its mistakes), the dyna-turtle game (exhibiting Newtonian laws of inertia), a curve-sketching program, and a few other goodies. Harvard Associates also mentions some attractions in coming updates to PC Logo, and allows the registered PC Logo owner one *free* update. Waterloo Logo provides a demonstration disk with graphics and music.

Of the four Logos, only DR Logo is copy-protected; however, Digital Research does provide one backup disk with its package. Extra backups cost $25.

The PCjr

As of this writing only IBM and PC Logo will run on the PC*jr*. Waterloo Microsystems has given assurances that by the time you read this its version also will run on Junior. Digital Research has said it has no immediate plans for Junior compatibility.

Conclusions

All of these Logos work really well, are lots of fun, and provide a good introduction to programming. All are easy to learn and easy to use, and all make use of the IBM keyboard and function keys. At $125, Waterloo Logo has the lowest list price, but that's not the whole story. Which Logo you buy may also be determined by what special features you need or want.

The following is a capsule summary, in alphabetical order.

DR Logo offers superior list-handling features and an extremely good user guide and reference manual. It also has a superior program-debugging facility. On the debit side, it has a minimal screen editor and runs under CP/M.

IBM Logo is fast, offers a complete set of enhancements, has an excellent reference manual and supplementary programming guide, and provides easy access to machine language programming (important for serious program development). On the other hand, it has a rather ordinary editor and is the most expensive of the four.

PC Logo has a great number of enhancements, particularly high-resolution graphics, 8087 support, and the ability to access both DOS and BIOS services. On a more elementary level, PC Logo has a user guide that's close to DR's in readability and thoroughness. But it's somewhat slower than the others at screen writing, it lacks a few list-handling capabilities, and its memory usage is nominally limited to 128K.

Waterloo Logo has the fewest enhancements, but a larger version is promised. Its small size and trim instruction book may make it appealing to people who don't need extras and who may already be familiar with Logo. Waterloo Logo is similar to Apple Logo; the Waterloo booklet, in fact, includes a short list of Waterloo Logo's deviations from Apple Logo. The presence of a *paint* command in Waterloo Logo is also a big plus for picture drawers, and its text editor is excellent.

The bottom-line good news is that you can purchase any of these Logos without fear of disappointment. ▲

```
TO HIL :COLOR
>SETBG 0 ; SETS BACKGROUND TO BLACK
>SETPAL 0; CHOOSES PALETTE 0
>SETPC :COLOR        ; SETS PEN COLOR TO INPUT :COLOR
>FULLSCREEN          ; USES WHOLE SCREEN FOR GRAPHICS
>HOME   ; SENDS TURTLE TO STANDARD POS.
>CLEARSCREEN         ; CLEARS SCREEN
>PENUP  ; TURTLE WON'T DRAW WHEN MOVED
>SETX 150; MOVE TO X=150, Y UNCHANGED
>SETY -100           ; MOVE TO Y= -100, X UNCHANGED
>PENDOWN ; TURTLE IS NOW IN LOWER RT.
           ; CORNER OF SCREEN,READY TO DRAW
>HILBERT 3 6 1       ; BIGGEST CURVE-FITTING SCREEN
END    ; PROCEDURE HIL ENDS

;THE NEXT PROCEDURE DRAWS THE CURVE. EACH HILBERT CURVE IS
;DRAWN BY DRAWING FOUR HILBERT CURVES OF A LOWER LEVEL WITH
;DIFFERENT ORIENTATIONS, CALLED PARITIES. NOTE HOW THIS
;PROCEDURE "CALLS ITSELF"—THIS IS RECURSION.

TO  HILBERT :SIZE :LEVEL :PARITY
>IF :LEVEL = 0 [STOP]      ; LEVEL 0 CURVES DO NOTHING
>LEFT :PARITY*90      ; ROTATE PARITY RIGHT ANGLES
>LOCAL "L; CREATE LOCAL VARIABLE L
>MAKE "L :LEVEL -1   ; SO YOU DON'T HAVE TO RECOMPUTE
>HILBERT :SIZE :L (-:PARITY) ; DRAW LOWER-ORDER CURVE
>FORWARD :SIZE        ; DRAW CONNECTING SEGMENT
>RIGHT :PARITY*90    ; CHANGE ORIENTATION AGAIN
>HILBERT :SIZE :L :PARITY    ; ANOTHER LOWER-ORDER CURVE
>FORWARD :SIZE        ; ANOTHER CONNECTOR
>HILBERT :SIZE :L :PARITY    ; SAME AS PREVIOUS
>RIGHT :PARITY*90    ; ANOTHER TURN
>FORWARD :SIZE        ; THE LAST CONNECTOR
>HILBERT :SIZE :L (-:PARITY) ; THE LAST CURVE
>LEFT :PARITY*90      ; THE LAST TURN
END     ; PROCEDURE HILBERT ENDS

;See TURTLE GEOMETRY (Abelson and diSessa, MIT Press) for
;this and other wonderful Turtle Graphics projects.
```

Listing 1. The Hilbert Curve.

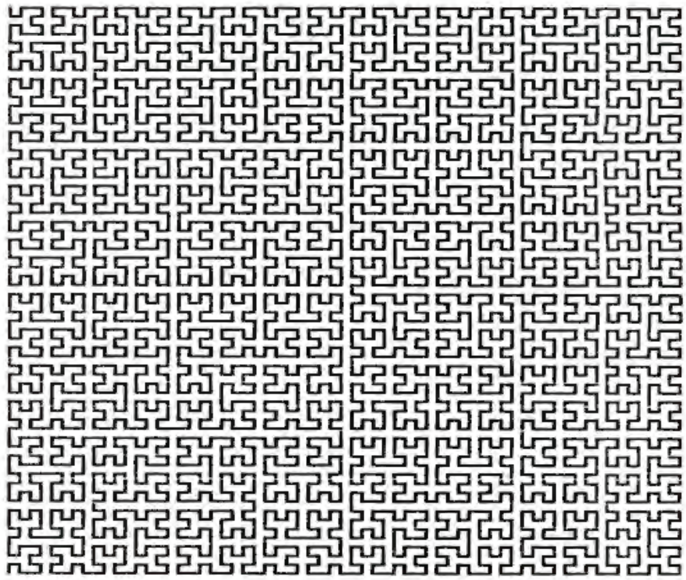
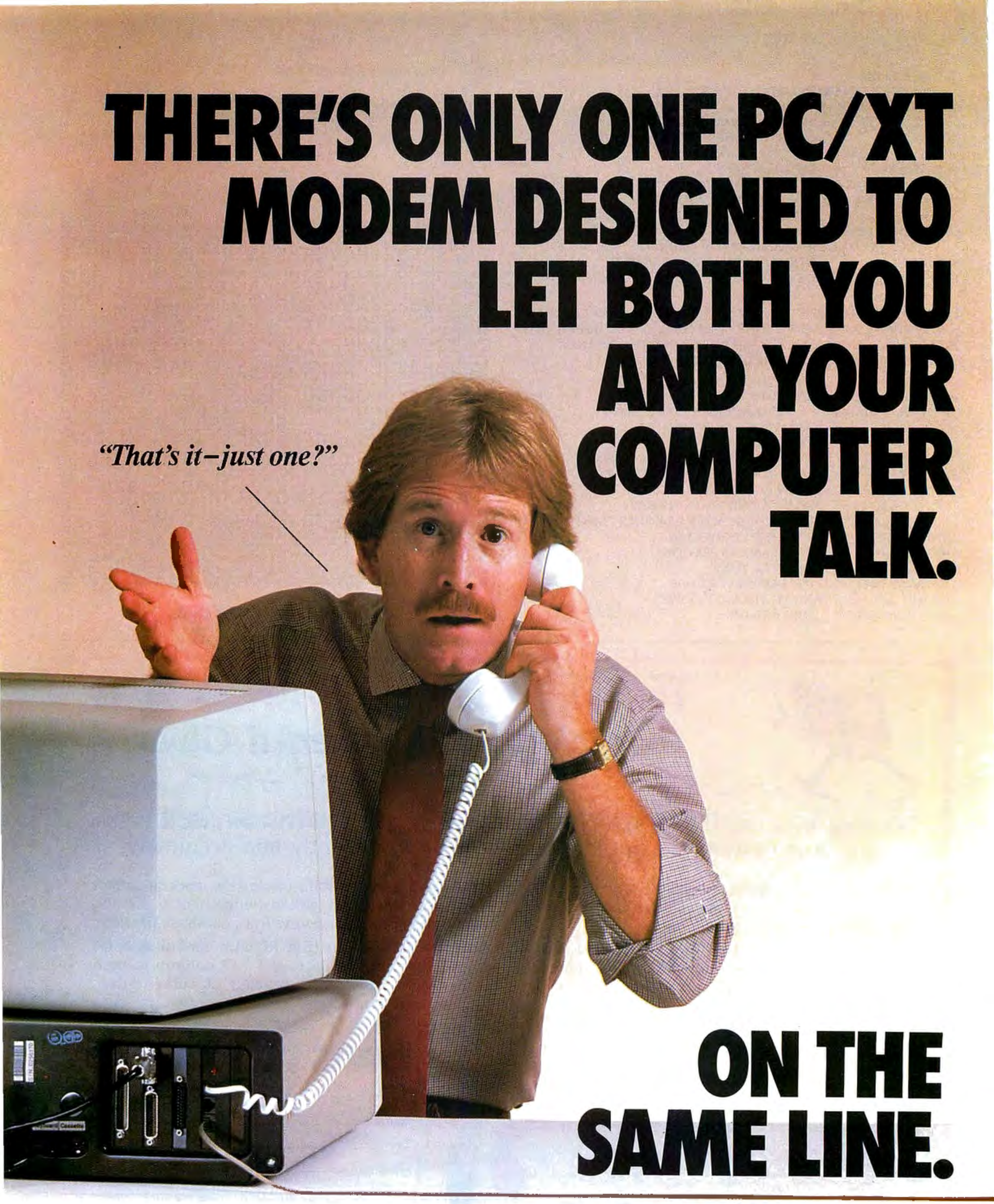


Figure 1. Level 6 Hilbert Curve.


THERE'S ONLY ONE PC/XT MODEM DESIGNED TO LET BOTH YOU AND YOUR COMPUTER TALK.
"That's it–just one?"
ON THE SAME LINE.

Only the PC:IntelliModem™ lets you switch repeatedly between voice and data communications.

Any old modem will let computers talk. But to move from talking or listening to sending and receiving data—at the touch of a single button—there's only one modem that's smart enough. The 1200-baud PC:IntelliModem from Bizcomp.

Let's say you want to transfer some files, and you want to talk to the person receiving them, both before and after the transfer. With other modems, you'd have to hang up, re-dial, or plug and unplug a bunch of cables. Not very convenient, especially if you use your phone a lot, and you have only one line.

With the PC:IntelliModem, you just plug your regular handset into the back of your IBM PC/XT. That's it. You're now ready to make connections with computers—and people—much faster and more conveniently. It's like getting a modem *and a telephone* for the price of a modem alone.

The ultimate status seeker.

The PC:IntelliModem also monitors your telephone line's status more closely than other modems. It listens to the line much like you do, and detects signals for dial tone, ringing, busy and voice—some of which other modems completely ignore. It then automatically takes the appropriate action, so there's less chance of error in making a connection. And since the PC: IntelliModem gives *you* a constant indication of what's going on, you always know how your call is progressing.

Make sure your modem has all these PC:IntelliModem features

Integrated Voice/Data
- Switch between voice and data communications
- Programmable telephone handset jack

Status Reporting
- Line status detection (dial tone, busy, ringing, voice answer, modem answer, incoming call)
- Audio monitor
- Programmable LED

PC:IntelliCom Software Included
- 99-name on-line telephone directory
- Auto-dial, auto-repeat dial, auto-answer
- Link to another number if busy
- File transfer
- Data capture to diskette
- Programmable auto log-on sequences

Compatible with Crosstalk™/PC-Talk III™

Receive Sensitivity: -50 dBm

Speeds: 110, 300, 1200 baud

A soft touch to operate.

Included with our single plug-in modem board is PC: IntelliCom,™ our exceptionally easy-to-use software package. It guides you step by step during use, through on-screen menus, entry instructions, an audible buzzer and a handy HELP key. There's even a demonstration program that runs through the main features.

All software tasks are activated by pressing function keys, which are clearly labeled on the screen. That means you can go from talking to someone, to transferring data, and back to talking again—all with single keystrokes.

The one that's out of sight.

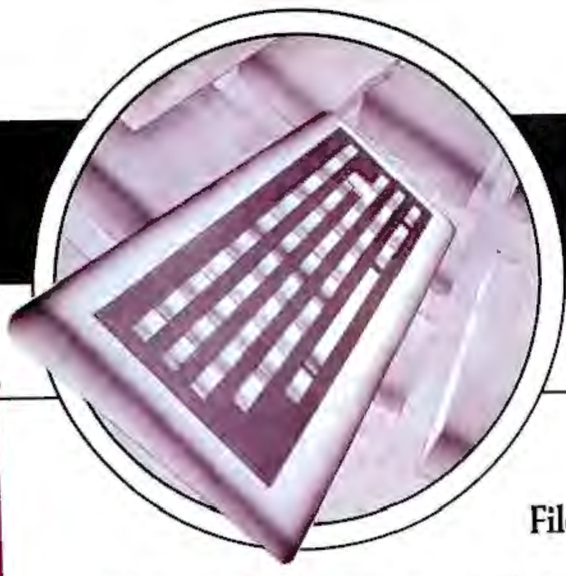
Even though the PC: IntelliModem is hidden inside your PC or XT, you should see how it works. So go to your dealer and ask for a demo. Also ask about our 2-year warranty, free NewsNet™ connect time and our modest $499 price.

Or contact:
Bizcomp,
532 Weddell Drive,
Sunnyvale, CA 94089;
408/745-1616.

You'll find there's only one modem that does it all, including voice communications: the PC:IntelliModem.

BIZCOMP®

We've got people talking.



BEGINNERS' CORNER

by Kathy Talley-Jones

Filenames and the *Copy* Command

Jane Barrett—who, as you'll remember, bought a PC*jr* with husband Kevin to make use of the myriad handy application programs (see *Softalk*, February 1984)—recently heard the call of her muse. She's decided to write a romance novel with the help of her PC*jr* and a word processing program. In writing her novel, she's made a useful discovery— namely, that some of DOS's housekeeping commands are faster and more comprehensive than the same features provided by her word processor.

You may remember from last month's discussion that DOS is the PC*jr*'s (and the PC's and the XT's) Disk Operating System—software that allows you and various kinds of application programs to work with the physical parts of the computer. Not only does DOS help manage application programs on disk, such as Jane's word processor, but it also ties in the peripheral devices. The keyboard, monitor, and printer are all examples of peripherals Jane will use in writing her novel.

Besides coordinating Junior's physical activities, DOS keeps track of separate files stored on floppy disks and helps Junior read and deal with what's in those files. For instance, the files stored on a working word processing disk (a working disk is one that you've copied from a distribution disk, and a distribution disk is one that you buy from a software vendor) might include Command .com, the DOS file that interprets your DOS commands; the file or files containing the word processing program itself; and files containing text that you've generated with the word processor. Jane's work disk contains chapters from her novel, *Oh, Wilderness Were Paradise Enow* (as in, "a flask of wine, a book of verse—and thou/Beside me singing in the wilderness. . .").

All these files obey certain rules that are laid down by DOS. For instance, the names of the files consist of up to eight characters, possibly followed by a period (sometimes pronounced *dot*) and a three-letter filename extension. The period and extension are optional; if present, the extension generally indicates the type of file—program file, backup file, text file, whatever. For example, it's customary to attach the extension .bas to files that store programs written in Basic (Basic, in fact, will supply that extension itself, unless you ask it not to—see this month's installment of "Basically Speaking"). Similarly, many application programs provide their own extensions—such as .bak or .wks.

Because Jane wants to be certain which files contain material for her novel—she doesn't want any steamy love scenes to get mixed into her letters home—she has given all the files for *Oh, Wilderness* the extension .nvl. DOS will recognize only one file by a given name and extension per disk, but you can have identically named files on other disks—backup disks, for instance.

DOS won't allow you to use certain charac-

Missing an installment of "Beginners' Corner"? The current column began in February 1984, but all back issues are still available; for further information, see page 4. The first "Beginners' Corner" columns (June 1982—September 1983) are now available as a single volume from Softalk Books.

ters in your filenames. The characters it permits are:

the letters A through Z
the numbers 0 through 9
the following symbols: ! @ # $ % ^ &
( ) - _ { } ' ` ~

You can use letters in either capital or lowercase form; DOS will treat them all as uppercase letters.

The characters that are off limits have special meanings to DOS. These characters are:

> < ¦ ? * / \ , .

A complete file specification includes, in addition to the filename and the extension, the disk drive identifier and pathname. Pathnames we'll take up later; on the PC*jr*, they're of little practical value. The disk drive specifier is another matter.

Since Junior has only one drive, you may think it odd that you need to concern yourself with specifying disk drives. However, the PC*jr* acts as if its lone disk drive were in fact two drives—drive A and drive B. That is to say, it treats its single *physical* drive conceptually as two *logical* drives. This schizoid behavior will come to your attention when you copy files from one disk to another.

To copy a file from one disk to another, issue a command of the form:

COPY NAME1.EXT B:NAME2.EXT

Let's take a moment to unravel this command. The word *copy* summons DOS's file-copying utility from memory. NAME1.EXT represents the name of the file you want to copy. The B: means you want to copy this file to another disk. And NAME2.EXT represents the name your file will have once it has been copied onto the new disk (this may or may not be the same as the original filename; we'll have more to say about this part of the command in a moment).

The B: is the drive specifier. It tells DOS that the copy you're making will go on the second logical disk drive; that's just another way of telling DOS that you want the original and copy to reside on separate disks. You could, by the way, have put an A: before NAME1.EXT in that copy command; but it's unnecessary, because as long as DOS is giving you A> as its prompt, it will carry out all your disk commands on drive A unless you tell it to do otherwise.

When you carry out the command shown above, DOS will start by reading your *source* file—the file you wish to copy. Then it will ask you to put your target disk in drive B. Your *target* disk is the disk on which you want the copy to live, and drive B is just the same disk drive by another name. When you've got your target disk mounted, you'll signal DOS to proceed by striking any key on the keyboard.

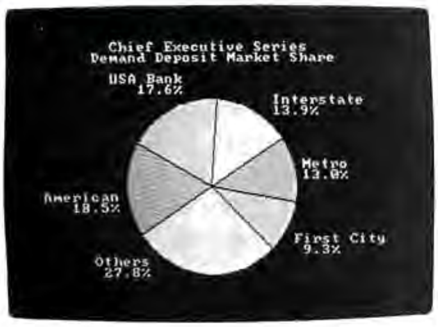
The moral of this tale is that if you want to copy a file from one disk to another, you need to include the drive specification in your DOS command.

You don't have to copy files onto separate disks, of course. So long as you provide a different filename or extension for your copy, and so long as there's enough room on your disk, you can make a copy of a file on the same disk that holds the original file—in other words, your source disk and target disk can be the same. To do this, simply follow the syntax of the *copy* command as shown above, but omit the B:. If you happen to forget and include the B:, it's no problem; when DOS prompts you to put in your target disk and strike a key, just strike a key. DOS can't tell whether you've swapped disks or not. If, however, you attempt to copy a file onto your original disk and fail to specify a new filename or extension for the copy, DOS will protest; remember, you can't have two files with the same name on the same disk.

The commands we looked at last month, *diskcopy* and *format*, are both *external* commands; external means you have to insert the DOS disk (or a copy of it) before you can use these commands. *Copy* is an *internal* command, which means you don't have to have the DOS disk in the drive to use it.

When Jane boots her word processing program, all the internal DOS commands (there

CHIEF EXECUTIVE SERIES



Introducing computer games that make you the C.E.O.

The Chief Executive Series is an exciting new concept in entertainment. Each game in the Series makes you the Chief Executive Officer of a company in a different industry. You make the decisions that are the key to success or failure. The quality of your decisions determines how well your company does against its competitors.

As you manage your company, you get vital information about the business environment, competitors, and your own performance, displayed in easy-to-understand bar charts, pie charts and graphs. You can play alone, against the computer, or in competition with others.

Chief Executive Series games are as educational as they are entertaining — an ideal way to learn the fundamentals of strategy and decision making in each of the industries represented. Every game gives you an authentic and realistic management challenge.

Chief Executive Series games are available for IBM and Apple computers. They will thoroughly test and sharpen your management skills.

BANK PRESIDENT You are president of a large commercial bank. You must plan your strategy and make all the important decisions involved in running your financial empire. Set loan and deposit interest rates, raise or lower employee salaries, issue and redeem stocks and bonds, and manage your investment portfolio. Available now.

HIGH-TECH ENTREPRENEUR Start and build a high-tech manufacturing company. You must organize a competent management team, raise venture capital, and successfully develop and market your products. To succeed, you have to develop good management skills and the ability to allocate limited resources. Available in April.

VENTURE CAPITALIST Compete with other venture capitalists for funds and for the best investment opportunities. Evaluate business plans and decide when and on what terms to invest. Provide seed money for start-ups or finance more established companies. An excellent way to learn more about the exciting venture capital world. Available in May.

Call Lewis Lee for the name of the dealer nearest you. 1-800-842-9900

LewisLee™ CORPORATION

P.O. Box 51831 • Palo Alto, CA 94303 • (415) 853-1220

are a number of others in addition to *copy*) are read into memory along with the file Command.com, a file that must be present on the disk from which DOS is booted. Ordinarily, these internal commands remain in your computer's memory until you turn the power off. Occasionally, if you're running a large application program, you may crowd them out of memory, in which case DOS will have to go back to disk to get Command.com again.

Jane wants to copy the chapter she's recently revised—in which she introduces the lovely Leticia and her husband-to-be, the faithless Frank—from her working disk onto her backup disk; she doesn't want her breathless prose to vanish because of some freak accident or because Kevin tripped over Junior's power box. Jane inserts her working disk, and, at the DOS prompt, A>, she types:

COPY CH1REV2.NVL B:

Junior responds:

Insert diskette in drive B and strike any key when ready

Jane removes her working disk, inserts the backup disk, and strikes the space bar—and DOS copies the file Ch1rev2.nvl onto her backup disk. Notice that Jane didn't specify any filename for the copy of her file; she just typed *B:*. DOS is no dunce; it realizes that Jane wants the file to have the same name on the target disk as it has on the source disk.

Jane can make sure that her precious words have been copied correctly by issuing the *comp* command, which compares the contents of one file against those of another. To use this command, she inserts the DOS disk (*comp* is an external command) and types

A>COMP

DOS replies:

Enter primary file name

At this point Jane must replace the DOS disk with the source disk (the disk on which her original file resides) and type

A:CH1REV2.NVL

DOS then reads the file and prompts:

Enter 2nd file name or drive id

Jane types

B:CH1REV2.NVL

or just

B:

DOS's response is:

A:CH1REV2.NVL
Insert diskette in drive B and strike any key when ready

Jane inserts the disk containing the copy of her file and strikes a key, and DOS says:

and B:CH1REV2.NVL

Files compare ok

Another way to see whether a file has reached its destination (although this method doesn't guarantee that the file's contents have been copied correctly) is to use the *dir* command to check the directory of drive B's disk. *Dir* is another internal command, so Jane can use it without inserting her DOS disk. She just types *dir b:*, and voila—

Volume in drive B is WILDERNESS
Directory of B:

CH1REV2	NVL	5888	4-20-84	7:15p
CH3REV3	NVL	1024	4-02-84	6:45p
CH5REV3	NVL	11776	3-30-84	2:28a
CH4	NVL	5760	4-16-84	8:19p
CH2REV3	NVL	11776	4-01-84	1:10a
PARENTS	LTR	5888	2-22-84	3:42p
CH2REV2	NVL	6369	3-29-84	11:31p

There's Ch1rev2.nvl right at the top of the list.

There's still another way that Jane can assure herself of the presence and integrity of her backup copy. She can use the /V (for *verify*) option of the *copy* command when she makes her backup in the first place. If she had typed

COPY CH1REV2.NVL B:/V

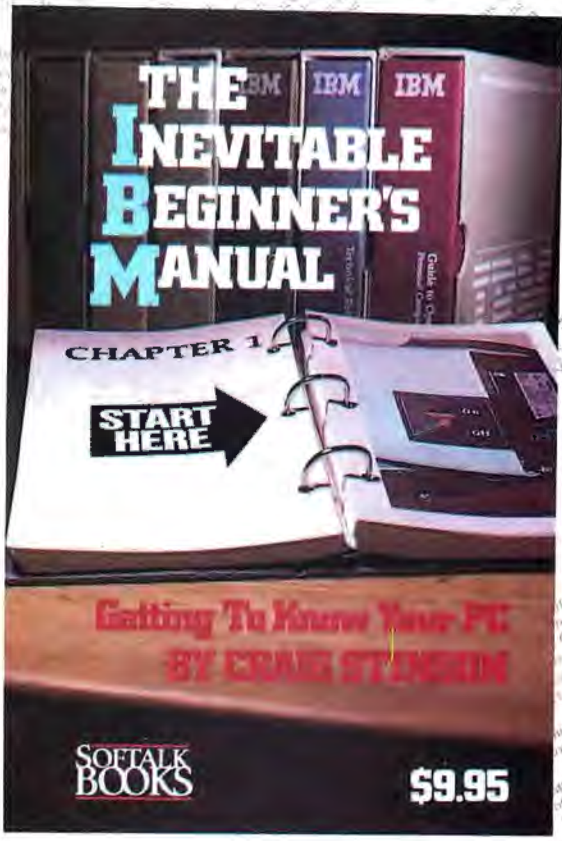
when she made her backup, DOS would have gone through an extra procedure to make sure the contents of the copy matched the contents of the original.

In revising her novel, Jane has created and erased a large number of files. Junior's disk drive head has to rattle all over the disk to read these files, because DOS doesn't always write files in adjacent sectors on a disk. Rather, it writes file sectors wherever it finds available disk sectors.

Jane has made a lot of erasures, so her files have become somewhat disjointed in their physical structure on the disk. (Any time you erase a file, DOS considers the sectors formerly occupied by the erased file to be available for new file storage; if you make lots of erasures and if the new files you create aren't exactly the same size as the ones you erase, then your files will become more and more physically disjunct.) Because her files are no longer stored in contiguous disk sectors, the time it takes Junior to read them into memory is slightly longer than it used to be.

If Jane wishes to make her disk drive read a little more efficiently, she can "straighten out" her files by copying them all onto a fresh disk—using, of course, the *copy* command. To make this task a little easier, Jane can take advantage of DOS's *wildcard* characters. We'll join her as she does so in the next installment of this column. ▲

INFORMATION



PLEASE

GET TO KNOW YOUR IBM PC BETTER

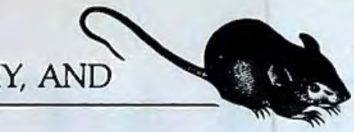
Discover more about what makes your computer work. Explore the possibilities of your machine. Let Craig Stinson take you on a guided tour.

Understanding the inner workings of your computer can help you to use it more efficiently. It can help get you out of situations heretofore requiring control-alt-del. This book offers explanations of the systems you interact with daily, and how to extract the best performance from them.

Use your computer intelligently. Let it do more. Find out more about it.

$9.95 ISBN 0-88701-004-0

Please add $1.50 for shipping/handling ($11.45 total).
California residents add 6.5 percent sales tax ($12.09 total).

The Origin, Anatomy, and  Varieties of *Mus computeralis*

by Tom Foth

It's the Year of the Mouse!

What's faster than a cursor key, more powerful than typematic, and able to point to screen areas with a single flick of the wrist? That's easy. It's a mouse for the PC, something that lots of PC users may find scurrying around the desktop this year.

Like the critters for which they're named, PC mice come in a variety of subspecies adapted to various environments. This article discusses the different kinds of mice you can get for your computer with capsule reviews of seven specimens. In the course of this article, we'll be looking at how mice are constructed, how they work, and how the different kinds compare. But first, let's look backward in time and survey the evolution of the computer mouse.

Birth of the Desktop Rodent

Like much of the other personal computing magic we're seeing today, the digital mouse has its roots in the Silicon Valley of the early 1960s. Back in those pre-micro days, a certain Dr. Douglas Engelbart, then affiliated with the Stanford Research Institute, had a vision: Computers should be programmed in such a way as to augment the human intellect.

Engelbart developed a computer system embodying these concepts, and he called it Augment. This system was a forerunner of today's integrated workstations—Xerox's Star and Apple's Lisa and Macintosh—and of integrated software packages, such as *1-2-3* and VisiOn.

In an effort to make his software fast and effective, Engelbart researched fundamental man-computer interactions. Since one of the basic ways that people use computers is through the manipulation of items on a screen, Engelbart reasoned that this was an appropriate area for experimentation. To that end, he investigated cursor-control and screen-selection devices such as light pens and joysticks. In an attempt to make his experimentation as complete as possible, Engelbart expanded on some of his previous ideas and constructed the first mouse. When he tested the mouse alongside other devices, he found that (to quote him directly) "it won over everything else, hands down."

The first mouse, circa 1964, was a two-by-three-by-four-inch block of wood with three buttons on top. Its bottom was hollowed out to make room for two wheels, each of which was attached to the shaft of a separate potentiometer. (A *potentiometer*, also called a *variable resistor*, is a device that controls the flow of electricity in accordance with the orientation of a shaft; volume controls on radios and televisions are examples of potentiometers.)

As described by Engelbart (writing with William English and Melvyn Berman) in the *IEEE Transactions on Human Factors in Electronics*, March 1967, the wheels of that

original mouse were mounted at right angles "to resolve the motion [of the mouse on a surface] into two orthogonal components in much the same manner as do the disks in planimeters or in the old-fashioned mechanical differential analyzers." An analog-to-digital converter translated the flow of electricity into digital signals.

In the early 1970s, Xerox's Palo Alto Research Center (PARC) set out to develop powerful personal computers for use in office automation research. The computer developed at PARC, the Alto, was the next major system to use a mouse.

Like the original mouse, the mouse created for the Alto had three buttons, and at first (in 1972), used Engelbart's right-angled wheel arrangement (although the potentiometers were replaced by a digital encoding system). This mouse, developed for Xerox by Jack Hawley (then simply a self-labeled "inventor") was supplanted by another in 1975; the latter, which used a large ball bearing, was manufactured by Hawley—who by then had become president of the newly formed Mouse House. Developed by Willard Opocensky of Xerox, the ball-bearing mouse, which was to become the standard in the mechanical mouse arena, was designed to do away with the drag caused when wheels were made to slide against their natural direction of travel.

Only fifty Altos were ever distributed outside of Xerox, and all of them went to the computer science research centers at Stanford, Carnegie-Mellon, and MIT. Even though few people ever saw, much less used, an Alto, the Alto's impact on personal computing will be felt for some time. Besides being the second significant mouse-using system, Alto brought icons to the personal computing environment. It is easy to see the Alto heritage of Apple's Lisa and Macintosh as well as in the Xerox Star.

Mouse Anatomy

There are two basic types of mice, mechanical and optical. Mice are categorized according to the way in which they translate mouse movement into computer signals. As mentioned earlier, the mechanical mouse is the older of the two types. It translates movement of mechanical components (caused by physical contact of the moving mouse along a surface) into digital signals for processing by a computer.

Mechanical Mice. Mechanical mice come in two basic varieties; the difference has to do with the way movement of the mouse is commuted to the motion-detection circuitry.

In the first group are mice based on Engelbart's design, mice that employ two wheels oriented perpendicularly to one another. As the mouse is moved about, each wheel turns an amount proportional to the vertical or horizontal component of the mouse's motion; in the special cases where the mouse is moved only horizontally or only vertically, one wheel is dragged while the other turns. While the idea of dragging a wheel across a desk surface may be unappealing, users of this type of mouse claim they have little problem getting angular accuracy.

The second group of mechanical mice are those that use balls or bearings to commute movement to the motion-detection circuitry; mice of this type dominate the mechanical pack. This design can be likened to an inverted track ball. As the mouse is moved about on a surface, the ball moves freely, transmitting the physical movement to—you guessed it—perpendicular wheels, which, in turn, drive the motion detectors. The function of the ball or bearing is largely to smooth out the movement of the mouse during angular travel.

Once the movement of the mechanical mouse has been resolved into orthogonal components (either directly via perpendicular wheels or indirectly by means of balls or bearings that drive perpendicular wheels), these components are translated into electrical signals by means of one of three types of motion detectors.

The first type, the one used by Engelbart's original mouse, is a simple potentiometer or variable resistor. (Since PC joysticks also use potentiometers for position determination, game adapters or similar analog-to-digital converters can be used to convert the analog resistance into digital signals.)

The second method of detection and conversion involves electrical contacts that make or break connection in response to the movement of a wheel. In one example of this type, the Mouse House Hawley X063X mouse, vertical and horizontal motion is resolved by two wheels and two sets

of four staggered wire fingers that alternately make and break contact as they wipe over the metal bands on the surfaces of the wheels. (See figure 1.)

The Microsoft Mouse uses a similar but more economical scheme, with metal arms wiping against alternating contacts on the sides of two wheels. (See figure 2.) As the wheels rotate, the circuitry inside this mouse determines which direction a wheel is turning by tracking the order in which the contacts make and break connection.

Encoding this motion into digital form is not as straightforward as it might seem. If the encoding wheel were constructed for simple binary encoding, situations would arise where both bits would need to change simultaneously—for example, the case where binary 01 changed to binary 10. And if one of the contacts ever lagged an instant in breaking or making contact, errors of a full binary magnitude could occur. For example, in the change from 01 to 10, a binary 11 might be registered momentarily to the interface hardware.

To circumvent this problem, a two-bit Gray code is used. A Gray code is one in which successively larger numbers are represented by single bit changes. In other words, the code is set up so that as numbers are incremented, two bits never change at once. Table 1 shows an example of a two-bit Gray code.

The Microsoft Mouse resolves mechanical motion directly into Gray code values. In the Hawley mouse, the four fingers trigger integrated circuits in the mouse (electrical gates called flip-flops) that in turn generate Gray codes.

The third motion-detection/signal-conversion method has been called a hybrid, because, like the true optical mouse (to be discussed later), it uses light-emitting diodes (LEDs) and light detectors to register motion. (See figure 3.)

On one side of this wheel is a light source (an LED), and on the other are two light detectors. As the wheel turns, the alternation of transparent and opaque material in the wheel causes the interruption and restoration of the light beam. For this reason, this type of motion detector is called a *light chopper.*

If the light chopper wheel turns clockwise, the light detectors see the pattern summarized on the left in table 2; if the wheel turns counterclockwise, they see the pattern shown on the right. In either case the pattern of ons and offs resembles that of the Gray code because both detectors never have to change values simultaneously.

The Wico mouse (used by Comco Concepts) and the Logimouse (from Logitech) use this method to detect motion.

Optical Mice. Optical mice sense movement by detecting the presence or absence of reflected light. They direct a beam of light against a grid; movement of the mouse over the grid results in patterns of alternating reflection and nonreflection. Because they have no internal moving parts (no wheels or ball bearings), optical mice don't require physical contact with a surface.

Two types of optical mice were reviewed for this article. Members of the first group, represented by Mouse Systems and Summagraphics, are fully optical; they use LEDs and light detectors to fully detect motion and direction. It is not difficult to understand why the methods used by these companies' mice are the same: The SummaMouse by Summagraphics makes use of patented technology licensed to them by Mouse Systems.

Both mice use a grid consisting of vertical and horizontal lines on a silver background. The paint used for the vertical lines is transparent to infrared light; the infrared light passes through the paint and reflects off the underlying silver surface. This same paint absorbs visible red light, thus preventing its reflection. The horizontal lines do the opposite; they absorb infrared light while allowing visible red light to reflect.

The mice consist of two light-emitting diodes (one generating visible red light, the other generating infrared) arranged so that reflections from the grid are picked up by one (Summagraphics) or two (Mouse Systems) four-quadrant light detectors. In each case, a microprocessor in the mouse controls the LEDs, senses reflections, and converts the reflection patterns into computer-readable information.

In the Summagraphics mouse, which has only one light detector, the microprocessor alternates between having the visible red light LED on and the infrared LED on. In both mice, the microprocessor detects motion by sensing reflections generated by either light source. (See figure 4.)

As shown in the figure, the light detector is sitting on a totally reflective silver strip (it's directly over neither the horizontal nor the vertical stripes of the grid); thus, light coming from either LED is reflected back to it. If the mouse is moved straight up from its position in figure 3, quadrants 1 and 2 of the four-quadrant light detector will pass over the infrared-absorbing horizontal

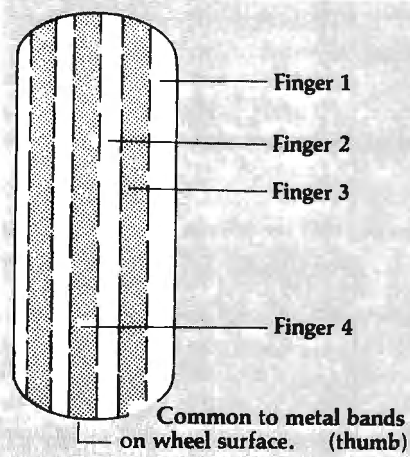


Figure 1.
The Mouse House Hawley X063X mouse (top of wheel surface).

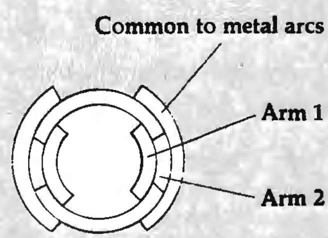


Figure 2. The Microsoft mouse.

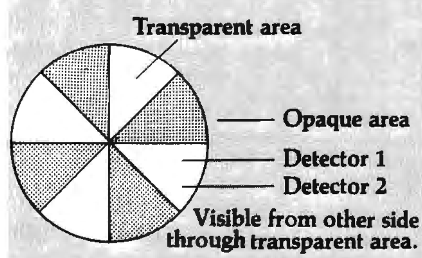


Figure 3.
Side view of light chopper wheel.

stripe and therefore stop receiving reflected light from the infrared LED. As the mouse continues to move in this direction, quadrants 3 and 4 will also cease to detect infrared light reflections. Then, as the mouse passes beyond the horizontal stripe, detectors 1 and 2 will start to sense reflected infrared light again, while detectors 3 and 4 remain in the dark with respect to the infrared LED. (See table 3).

By comparing table 3 with table 2, you can see that the Mouse Systems/Summagraphics scheme results in resolution of motion into Gray code. Horizontal motion is detected via reflections of light from the visible red LED. Motion in this direction is detected by light-detector pairs 1-3 and 2-4. Thus, it's easy to see that the LED/light-detector arrangement can detect direction as well as amount of motion.

The second optical type, represented by the USI mouse, is in many ways a hybrid. Like the Mouse Systems/Summagraphics mice, the USI mouse detects motion by using an LED/light-detector arrangement.

Unlike the Mouse Systems/Summagraphics mouse, however, the USI mouse employs only one LED with a uniary light detector on a surface of purely reflective and nonreflective horizontal and vertical lines. Since a uniary light detector cannot convert motion into Gray code (the whole detector senses light or doesn't), the USI mouse uses an LED/light-detector combination to register distance only. As the light detector alternately senses reflected light and darkness, the microprocessor in the mouse counts the number of lines it is passing over.

To detect the direction of motion, the USI mouse uses a novel optical switch. This switch is composed of a reflective metal post that is connected to the free-floating bottom of the mouse and two LED/light-detector pairs. As the post is urged by the movement of the mouse toward or away from either of the LED/light-detector pairs, light is respectively reflected or not reflected back to the detector (see figure 5). Thus in upward movement, the post is pushed toward LED/light-detector pair 1, with the result that light is reflected and the upward motion is detected. Similarly, if the mouse is moved to the left, the post is directed toward the right, reflecting light back to the light detector in pair 2 and thereby signaling leftward movement. If no light is detected by the detector in pair 1, that means the mouse is being moved down: If no light is being detected by the detector in pair 2, the mouse is being moved to the right.

The Best Type of Mouse

While it may be that PC owners are ready to welcome the arrival of mice on their desktops, there's no general agreement among them as to the best type of mouse. Whenever that topic comes up in a group of mice users, it seems there are as many opinions on the matter as there are mice. Fortunately the controversy resolves ultimately to a single question: Which is better, the optical mouse or the mechanical?

The advocates of mechanical mice consider the optical mouse's need for a reflect grid to be a liability. They claim that the mechanical mouse runs on any flat surface. While for the most part their claim is valid, some mechanical mice do skid on slippery desk surfaces. A coarser surface, such as that provided by a sheet of bond paper, seems to eliminate this problem.

Optical mice advocates consider the mechanical mouse inferior because it's mechanical. They reason that all those moving parts create too many opportunities for a part to break or become jammed. Further, the tracking balls on mechanical mice pick up grit from the desk that can later gum up their machinery. To counter this problem, one maker of mechanical mice, the Mouse House, has incorporated self-cleaning rings to keep the ball clean. Others provide a simple method of removing the ball for cleaning.

So what's the best mouse? That's hard to say, because the question raises issues of personal taste. Mice should be chosen on the basis of the application(s) they'll be put to, the environment(s) in which they'll be used, and the most appropriate hardware interfaces for the application/environment.

Maybe if the optical mice proponents could get desk manufacturers to build in optical grids. . . .

Interfacing a Mouse to Your PC

For a mouse to serve as anything other than a high-tech paperweight, it must be able to inform its computer host that it has been moved. Conveying such information is the job of the mouse-computer interface circuitry. Several kinds of interfaces are possible.

The Game Adapter. The first type, one that is not currently used by any PC mice vendors, is the analog-to-digital converter in the form of a PC game adapter. When a game adapter is used in conjunction with a

Decimal	Binary	Gray
0	00	00
1	01	01
2	10	11
3	11	10

Table 1. Binary coding and Gray coding.

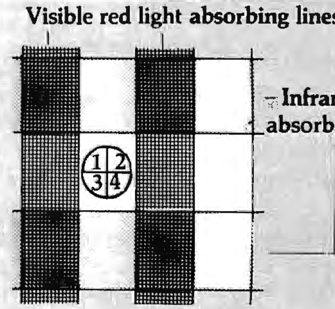


Figure 4.
Top view representing detector on grid.

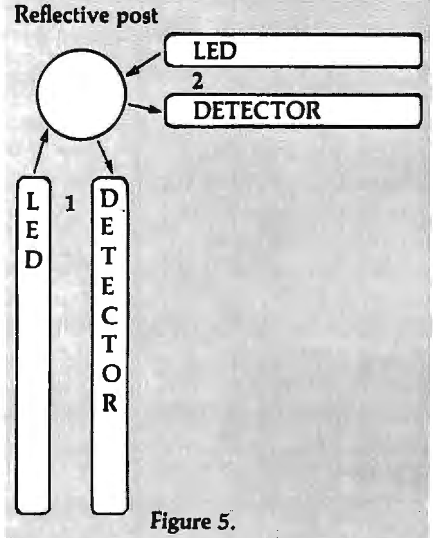


Figure 5.
Top view representing reflective light post with adjacent LED/light detector devices.

Detector 1	Detector 2
on	on
off	on
off	off
on	off
on	on
pattern repeats. . .	

Clockwise motion of light chopper wheel.

Detector 1	Detector 2
on	on
on	off
off	off
off	on
on	on
pattern repeats. . .	

Counterclockwise motion of light chopper wheel.

Table 2.

Detector	1 & 2	3 & 4
	on	on
	off	on
	off	off
	on	off

Upward movement of Summagraphics/Mouse Systems mice.

Detector	1 & 2	3 & 4
	on	on
	on	off
	off	off
	off	on

Downward movement of Summagraphics/Mouse Systems mice.

Table 3.

mouse, the mouse substitutes for one of the joysticks.

The advantages of interfacing via game adapter are that some systems already have one (and thus the cost of attaching a mouse is reduced) and that the BASIC *stick* function can be used to read the mouse position.

There are some disadvantages to this interface, however. One is that the programmer has to poll the game adapter; the adapter lacks the ability to interrupt a program and say when one of the analog signals has changed. Another disadvantage is that the PC memory map allows for only a single game adapter (and that adapter's I/O address is not configurable). Thus the mouse will tie up at least one joystick connection (if the mouse vendor supplies the proper cable set) or, worse, both joystick connections (if the mouse plugs directly into the game adapter). Furthermore, if the mouse is to be used on a PC that doesn't already have a game adapter, an adapter will need to be purchased and installed, and another PC slot will be consumed.

RS-232. The next type of interface is through the RS-232 port. Probably the most popular choice, this method is used by Mouse Systems, Summagraphics, USI, and (optionally) by Microsoft and Random Access (when the Random Access interface is used in conjunction with the Hawley mouse). Using this interface, PC software communicates with the microprocessor inside the mouse by issuing commands to and reading responses from the mouse over one of the PC-DOS COMn: devices. Unfortunately, there's no standard mouse language; each mouse has its own command and response syntax. Some mice, however, can emulate other positioning devices such as bit pads.

One of the principal advantages of the RS-232 interface is that, since most PCs already have at least one RS-232 port, the likelihood is that no additional cards will be required. To communicate with the mouse, PC software simply uses the COMn: device support provided in most PC languages. Since these devices report status changes to the computer automatically, it's not necessary for the programmer to poll the RS-232 interface. Finally, the mouse can be unplugged and moved to another system (PC or not) that has an RS-232 port.

The primary disadvantage is that for PCs with only one RS-232 port and applications that require some RS-232 device to be used concurrently with an RS-232 mouse, another port will be needed (and, probably, another slot will be used). This situation could arise, for example, if the mouse is used with a terminal-emulation program or concurrently with a serially interfaced printer. A final disadvantage is that most mice interfaced this way require an additional power supply (of the calculator recharger type).

(Note: At the time of this writing, Microsoft was not going to document the protocol used in their serially interfaced mouse, and they were going to require applications to use either their special interface software or Microsoft Windows. This requirement would prevent use of their mouse on non-MS-DOS systems.)

The Keyboard. A third approach, one taken by Comco Concepts, Logitech, and Corman Custom Electronics (in conjunction with the Hawley mouse) is to interface the mouse at the PC's keyboard connection. When this is done, the mouse interface plugs into the keyboard socket on the back of the PC and the keyboard plugs into the interface.

The obvious major advantage of the keyboard interface is that it requires no additional cards and thus doesn't tie up any slots. Furthermore, since the PC can't tell the difference between a character typed at the keyboard and one it receives through the mouse interface, programs need only to read from the keyboard to detect mouse movement. And, just like characters produced at the keyboard, characters produced by the mouse and delivered through this interface can generate interrupts. Finally, this type of interface doesn't require an external power supply.

Since there's no standard in the area of keyboard interfaces, the major disadvantage to this method is that it makes the mouse less readily transportable from system to system than the RS-232-interfaced (or even the game-adapter-interfaced) mouse. For example, since the PC*jr* doesn't have the same keyboard connector as the PC, you couldn't use the same mouse interface on both machines. Some vendors do make different models of their interfaces for different types of PCs, however. Still, if one mouse is to be used with different types of PCs, an adapter will be needed for each different system type.

The Dedicated Board. The final variety of adapter is the dedicated board adapter, offered optionally by both Random Access (for use with the Hawley mouse) and by Microsoft. As its name implies, this interface is a special single-purpose logic board.

Since it's directly attached to the bus, this board has the advantage of speed (it doesn't employ an RS-232 or keyboard interface as "middleman"). It doesn't tie up game adapters or RS-232 ports. And it does provide an interrupt (although it's simply a timed interrupt, not one that's linked to mouse movement). Since dedicated mouse-interface boards "borrow" power from the PC bus, they do not require external power supplies.

This interface shares with some of the previously discussed interfaces the disadvantage of being less than transportable; the adapter can be used only on systems for which it was designed. If the mouse is to be shared among systems of the same type, each system must have its own adapter (or the system can be opened and the interface can be swapped from system to system). If a mouse is to be shared among dissimilar computers, each unique system will need a unique interface.

There are some other disadvantages to the special-purpose board. The interface ties up a PC slot with a single-purpose card. Since the mouse is plugged into the bus, care must be taken that the mouse interface address and the interrupt it generates do not cause contention with any other add-on boards in the system. And, since the interface doesn't use existing "routes" into the system, special interface software routines are required.

Interface Software

Once a mouse has been connected to a PC, the next challenge is to write a program to work with the mouse. There are three levels of interface software for mice: system-managed interfaces, transparent interfaces, and application-managed interfaces.

System-Managed Interfaces. Digital Research's *Concurrent CP/M-86* with windows, Microsoft's Windows, Quantum Software Systems's *QNX*, and VisiCorp's VisiOn are examples of system-managed interfaces. In these environments, the mouse is an integral part of the operating system, and applications that run in these environments can make use of the mouse through standard operating system calls. In addition, some of these systems provide mouse-to-cursor translations for applications that understand cursor movements but aren't fully integrated into the operating system (see transparent interfaces, next page).

It's worth pointing out that all examples of this type of software interface work only with specific brands of mice; in other

Manufacturer: The Mouse House, 1741 Eighth Street, Berkeley, CA 94710; (415) 525-5533
Product name: Hawley X063X
Technology: Mechanical (rolling metal ball/rotating wheels with contacts)
Resolution: 200 parts per inch
Number of buttons: 3
Interface: PC card or RS-232 (supplied by Random Access, 246 Highland Road, Pittsburgh, PA 15235; (412) 247-7472); or keyboard (supplied by Corman Custom Electronics, 38 Bridgeport Road East, Waterloo, Ontario, Canada N2J 2J5; (519) 844-4430). The keyboard is also available from Comco Concepts (see Keyboard Mouse review).
Software: Examples are supplied with Random Access MM-1 PC card.

A key download routine and example software for downloading the interface is supplied with the Corman PC Mouse Trap. The PC Mouse Trap is supported by Quantum Software Systems's Qunix (QNX) and Watsoft's Port operating systems.

Other features: The Random Access PC card interface uses eight consecutive I/O addresses that are switch-selectable. Furthermore, the interface provides a selectable timed interrupt for use by the PC software.

The Random Access RS-232 interface can be configured for any of eight combinations of word size, parity, and number of stop bits. It can be configured to send status information on demand of the application program—periodically, continuously, or when the mouse's state changes. Various data-presentation options are available. The interface will run at speeds from 50 to 19,200 baud. All operating parameters are dynamically alterable.

The Corman PC Mouse Trap can be downloaded to simulate three keystrokes (one for each button). Shift, alt, and control keys are not allowed. It can also operate in a graphics mode, where the interface sends an escape sequence when the mouse has moved a user-specified distance. The cursor resolution can be downloaded.

For a description of the Comco Concepts interface, see Keyboard Mouse review).

Comments: The Hawley X063X mouse is probably the finest mechanical mouse available. The ball-bearing assembly that supports the ball that engages the surface provides the smoothest tracking of all of the mechanical mice reviewed. The electrical contacts on the commutator are gold-plated. Finally, the self-cleaning ring arrangement solves most of the problems that can arise from the accumulation of dirt in a mechanical mouse. It's obvious that this mouse is the product of nine years of evolution.

The Corman PC Mouse Trap can generate cursor sequences only for movement and thus is not compatible with unmodified versions of *WordStar* (and other packages that use nonstandard cursor sequences). It doesn't have nonvolatile memory, so downloaded keystrokes must be reprogrammed each time the machine is turned on (as in the case of the Logimouse, this can be done in a PC-DOS batch file). Software is provided for PC-DOS; if the mouse needs downloading before being used with other operating systems, you need to run PC-DOS first to accomplish the downloading.

Price: Hawley X063X mouse, $300; Random Access MM-1 PC card interface, $295; Random Access MU-1 RS-232 interface, $295; Corman PC Mouse Trap keyboard interface, $189; Comco Concepts keyboard interface, $229

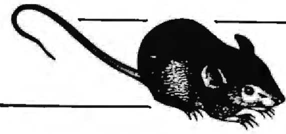
Manufacturer: Microsoft Corporation, 10700 Northup Way, Bellevue, WA 98004; (206) 828-8080
Product name: Microsoft Mouse
Technology: Mechanical (rolling metal ball/Gray-coded contact wheels)
Resolution: 100 parts per inch
Number of buttons: 2
Interface: PC card or RS-232 interface
Software: *Notepad* full-screen editor; *Game of Life;* sample piano simulation; *Doodle* drawing package; *Makemenu* pop-up menu utilities with examples for *WordStar, VisiCalc, 1-2-3,* and *Multiplan;* device drivers for both PC-DOS 1.1 and 2.0; application subroutine library.
Other features: The Microsoft Mouse is fully compatible with Microsoft *Word* and Microsoft Windows.

You can remove the ball for cleaning by loosening one screw.

Comments: The only flaw in this mouse is that it doesn't seem to move as freely as the Mouse House (Hawley) or Logitech mice. (The interface is compatible with the Mouse House [Hawley] mouse, allowing users to substitute that mouse.) It is easy to handle.

At the time of this writing, Microsoft was not going to document the protocol it uses with the RS-232 interface. Failure to do this would prevent the use of this mouse by applications that don't have access to MS-DOS and the supplied driver and subroutine library. The PC card version of the interface is not documented either and thus cannot be used in non-MS-DOS environments (such as CP/M-86 and PC/IX).

To its credit, the Microsoft Mouse appears to be the center of a possible standard in mouse software interfacing. Furthermore, the software supplied with the mouse is alone worth the cost of the mouse. *Notepad* is a very adequate full-screen editor, and *Makemenu* greatly extends the utility of the mouse; the ability to set up customized pop-up menus for any application is a very powerful feature. *Makemenu* is capable of generating nonstandard cursor sequences (so it can be used with packages such as *WordStar*), and pop-up menus provide virtually limitless keystroke-sequence generation.
Price: $195 (either version)



Manufacturer: Mouse Systems Corporation, 2336H Walsh Avenue, Santa Clara, CA 95051; (408) 988-0211
Product name: PC Mouse
Technology: Fully optical
Resolution: 100 parts per inch
Number of buttons: 3
Interface: RS-232
Software: Pop-up menu software for the following: *1-2-3, Volkswriter, WordStar, VisiCalc, Multiplan, Personal Editor;* Microsoft-mouse-compatible drivers and subroutine library.

Optional software includes the *MouseWindow* graphics development package for PC-DOS and either the Hercules or the IBM color/graphics card.

Other features: The baud rate can be set to either 1200 or 2400 baud. Communications protocol is eight data bits with no parity. Communication with the mouse takes place when there is a change in mouse state (motion or button change). Mouse Systems Five-Byte Relative protocol is transmitted to the host to communicate state changes (for other protocols, contact the factory).

Comments: The PC Mouse has nice handling characteristics. It's comfortable to hold and quick to respond. The pop-up menu package is well done: Its value will be enhanced when the user-customizable software is provided by Mouse Systems. The Microsoft compatibility makes the PC Mouse compatible with *Word* and, presumably, with Windows. VisiCorp has announced that VisiOn will support the PC Mouse.

Like the Microsoft mouse *Makemenu* package, the PC Mouse pop-up menus allow the mouse to generate nonstandard cursor sequences as well as giving it the ability to generate virtually unlimited keystroke sequences.

Price: PC Mouse, $295; *MouseWindows*, $69

Manufacturer: USI Computer Products, 71 Park Lane, Brisbane, CA 94005; (415) 468-4900
Product name: OptoMouse
Technology: Hybrid—optical/floating motion plate
Resolution: 100 parts per inch (greater resolutions are available)
Number of buttons: 4 (2 on each of 2 rocker switches)
Interface: RS-232
Software: None
Other features: The interface is capable of providing a variety of communications protocols (word size, parity, number of stop bits). Baud rates supported range from 300 to 19,200 baud and can be jumper-selected or automatically detected by the mouse. The mouse can emulate protocols of the following devices: Summagraphics Bit Pad One, binary; Summagraphics Bit Pad One, ASCII; Houston Instruments Hipad; Mouse Systems, Five-Byte Relative; Tektronics Plot 10; Retrographics RG-512 as well as generic three-byte relative.

Furthermore, most mouse parameters (resolution, axis orientation, transmit interval, and auto-scaling) are dynamically alterable. Mouse position is transmitted to the host on interval, on demand by the host, or on mouse motion (all of which are dynamically selectable).

Comments: Access to four buttons may be useful for many applications. The mouse has excellent handling characteristics. The floating base plate (used to detect direction of motion) results in a barely perceptible drag, however. The base plate can easily be removed (two screws) for cleaning.

This is probably a "best buy" for systems integrators who plan on incorporating a mouse into existing packages that already employ one of the USI-supported protocols.

Price: $299

Manufacturer: Summagraphics, 35 Brentwood Avenue, Box 781, Fairfield, CT 06430; (203) 384-1344
Product name: SummaMouse
Technology: Fully optical
Resolution: 100 parts per inch
Number of buttons: 3
Interface: RS-232
Software: Sample software
Other features: The SummaMouse is capable of baud rates ranging from 300 to 19,200. Baud rates can be automatically chosen by the mouse or set dynamically. The mouse can send data on change in mouse state or it can be polled by the host. The SummaMouse supports the following protocols: Summagraphics MM Series Digitizer (standard); Summagraphics Bit Pad One, binary (special-order from the factory); Mouse Systems, Five-Byte Relative (special-order from the factory); Generic Three Byte Relative (special-order from the factory).

Comments: Summagraphics licensed the SummaMouse technology from Mouse Systems. The SummaMouse is very similar in appearance to the Mouse Systems PC Mouse.

A retail version of the SummaMouse was not available at the time of this review but should be available at the time this article appears. Specifications are subject to change.

This may be a "best buy" for systems integrators and developers. It is Summagraphics's intent to standardize on one protocol for their mice, bit pads, and digitizers.

Price: $300 (subject to change)

words, specific systems require mice from specific manufacturers. VisiOn takes this limitation one step further: Only one particular mouse will run with a unique copy of VisiOn. During initialization, VisiOn interrogates the mouse (a custom SummaMouse by Summagraphics) by serial number to see if the particular copy of VisiOn recognizes the particular mouse. If the mouse does not present the serial number encoded into the VisiOn software, the software doesn't run.

Transparent Interfaces. At the transparent interface level, we find such software as Trillian's *visuALL*, Quarterdeck's *Desq*, Microsoft's *Makemenu* (which comes with the Microsoft Mouse and is available as an at-media/postage-cost upgrade to current owners) and PopUp Menus (which comes with the Mouse Systems PC Mouse). Also, mice interfaced through the keyboard (Comco Concept's Keyboard Mouse, Corman Custom Electronics's PC Mouse Trap with the Hawley mouse, and Logitech's Logimouse) provide a high degree of transparency. When used with this software or these mice, PC applications are not aware that a mouse exists on the system. The software or mice appear to the program as if someone were typing at the keyboard.

In addition to translating mouse movement and button presses into their keyboard equivalents, most of these software interfaces provide pop-up menu environments similar to those of VisiOn and Apple's Lisa, with menus for popular packages (*WordStar, 1-2-3, Multiplan, VisiCalc*) and a menu-development language that allows users to develop their own menus. As is the case with system-managed software interfaces, only certain vendors' mice work with certain transparent interface software.

Application-Managed Interfaces. At the lowest level are those software interfaces that require the application to communicate with the mouse specifically, either through a set of subroutines (like those available with the Microsoft Mouse) or through a COMn: file. The major disadvantage of this type of interface is that it can't be used with packages that have no awareness of the mouse. Countering that disadvantage is the fact that for new applications this kind of interface provides the most control over the mouse.

The Future of the Species

Is the mouse just another fad? Most mouse vendors think not. David Pruner, marketing director of Summagraphics, feels that

mice represent the beginning of a whole new evolution of graphics input devices. In his view, if a manufacturer could make one cheaply enough, a mechanical mouse could replace the joysticks now found on video games. Pruner also believes that the optical mouse will dominate the market as the premier pointing device and that the digitizing tablet will become the major high-precision and high-function general graphics input device.

Dr. Engelbart, who started this whole evolutionary development, feels that while the mouse, when used with a conventional keyboard, may have mass appeal, it is only part of the optimal man-machine interface. He points to another man-machine interface that came out of the early Augment research: the one-handed chording keyset. This keyset, consisting of five keys (which resemble five white keys on a piano) is "played" by the user's left hand, much as a musical keyboard would be played, while the user's right hand works a mouse. By "chording" the five keys in different combinations, the user can achieve thirty-one different states (chords). Further, when used in conjunction with the three keys on a mouse, the user can choose one of seven chord sets giving him or her access to as many as 217 different symbols. The keyset would not replace the conventional keyboard; indeed, research has shown that for more than ten characters it's faster to use a conventional keyboard. When the keyset is used in combination with the mouse for entering commands and short literals, however, it allows operators to keep their eyes on the screen and saves them the time that would be required to move their hands from the mouse to the keyboard and back again.

Engelbart believes that keyboards, mice, and keysets are only part of what will be a much fuller environment once voice recognition has become a viable option. For the tighter the man-machine interface, he reasons, the more the computer can augment the human intellect. ▲

Manufacturer: Logitech, 165 University Avenue, Palo Alto, CA 94301; (415) 326-3885
Product name: Logimouse and Logimate for the IBM PC
Technology: Mechanical (rolling metal ball/light chopper)
Resolution: 200 parts per inch (320 parts per inch also available)
Number of buttons: 3
Interface: Keyboard
Software: Logimate PC-DOS command; Logimate.dat (PC-DOS file with definitions for *SuperCalc* and *1-2-3*).

Other features: The mouse is programmed by execution of a PC-DOS command that specifies cursor resolution and three eight-keystroke strings (each of which you can generate subsequently by depressing the appropriate mouse button). Mouse settings can be stored in a configuration file and called by name.

The ball is easily removed for cleaning.

Comments: One drawback of this mouse is that the cursor sequences are not programmable (thus preventing its use with packages, such as *WordStar*, that employ nonstandard cursor sequences). The download software provided is for PC-DOS only (of course you can use PC-DOS to load the mouse and then boot another system). There is no nonvolatile memory with which to back up the keystroke sequences. This is not an inconvenience, however, since the Logimate command can be called from a PC-DOS batch file prior to execution of an application.

The mouse is of very high quality and tracks smoothly. While it is larger than most others reviewed, it is easy to handle.

Price: $268

Manufacturer: Comco Concepts, 465 Convention Way, Redwood City, CA 94063; (415) 364-3464
Product name: Keyboard Mouse
Technology: Mechanical (rolling metal ball/light chopper)
Resolution: 24 parts per inch
Number of buttons: 3
(The above pertains to the optional Wico mouse. This interface is available alone for use with the Mouse House [Hawley] mouse.)
Interface: Keyboard
Software: None (interface software is internal to the interface)

Other features: The mouse can be configured for three different sets of cursor and button sequences. These sets are stored in the interface's electrical, alterable read-only memory (so they're nonvolatile); you select them by pressing a button on the mouse and typing two characters at the keyboard. You can store four five-character strings and call them up by pressing buttons on the mouse. The interface is factory-programmed for *WordStar*, *SuperCalc*, and *1-2-3*. The mouse's resolution is also user-selectable and can be saved in each of the three sets.

Lowercase and control characters can be programmed. Uppercase and alt characters are not permitted.

Comments: The interface can be purchased with or without the Wico mouse. The Wico mouse does not track as smoothly as the other mice reviewed. Furthermore, it has a large profile that could become burdensome if held by small-handed persons for long periods of time. Finally, there is no simple way to remove the ball for cleaning.

The interface can be custom ordered from the factory for use with the Hawley mouse (which must be ordered separately).

Configuring the mouse is straightforward; the user simply enters the keystrokes to be emulated in response to a menu presented by the interface. This saves the user the trouble of consulting a chart of scan codes. Since nonstandard cursor sequences can be programmed, the interface can be used without modification with packages (like *WordStar*) that employ nonstandard cursor sequences. The fact that you can't program uppercase and alt-character sequences is a limitation (attempting to enter these sequences causes the interface to fail), as is the fact that only three key-sequence sets are retained in nonvolatile memory.

Price: Interface alone (specify Wico or Hawley interface), $229; Interface with Wico mouse, $299.

Interfaces are also available for Columbia, Fortune, and Texas Instruments personal computers. A special RS-232 version for use with terminals is also available.

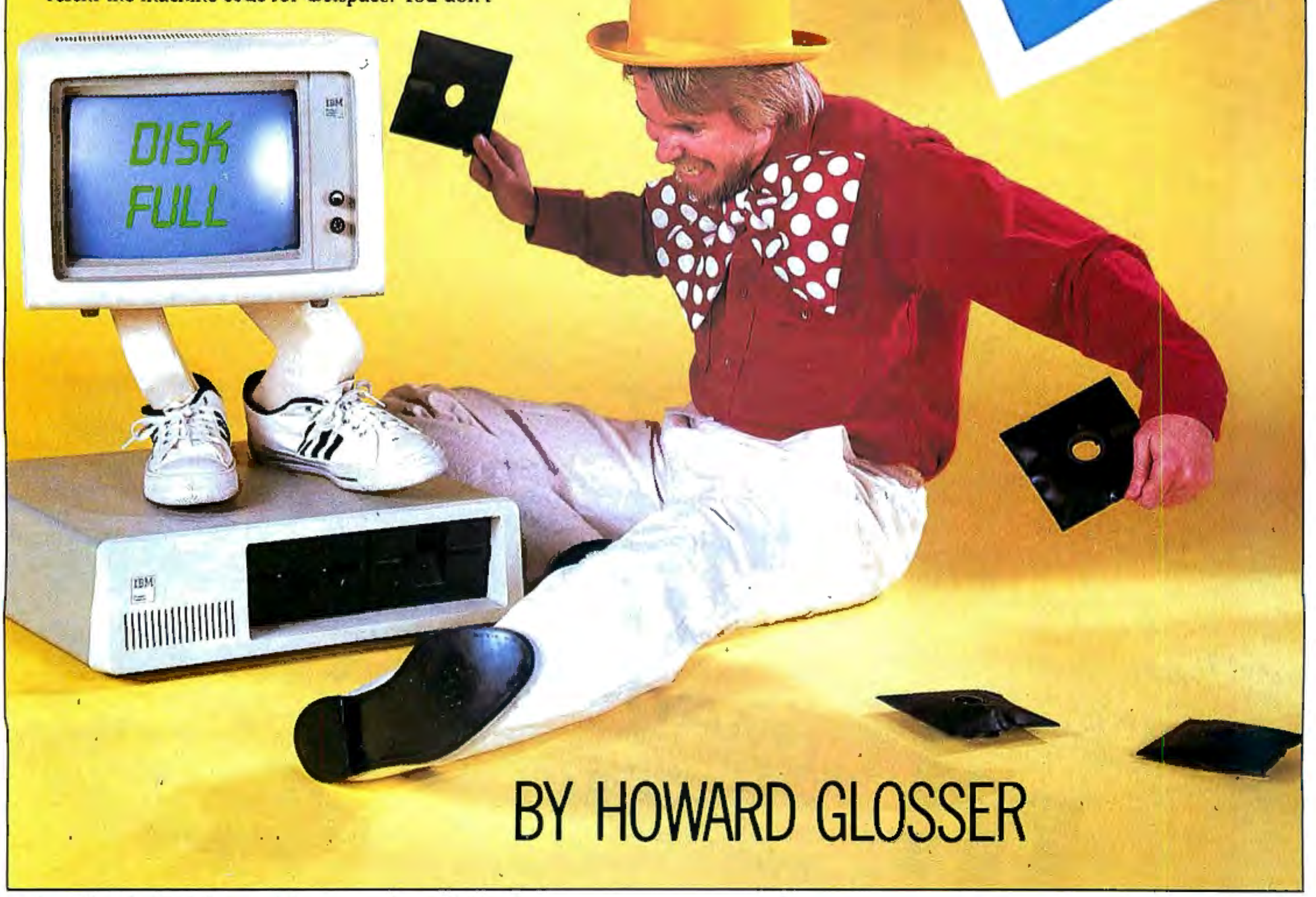
THE BASIC ASSEMBLY LINE

The GETSPACE Routine

When your BASIC program needs to write to a disk, wouldn't it be nice to know for sure that enough space is available on that disk? Many times, the infamous *Disk full* message appears on the screen after part of the output has been written. At that point, unless the program catches this type of error and gives you another chance, you must rerun the program with another disk that (you hope) has enough space available.

This month's subroutine, called Getspace, helps solve this problem. Although little can be done to remedy this situation in programs that you use now, when you write BASIC programs in the future with Getspace, you'll be able to tell exactly how many bytes are available on a floppy or hard disk before writing an output file.

The subroutine is built by a BASIC program containing data statements that represent the machine code for Getspace. You don't



BY HOWARD GLOSSER

NOW! Choose From 3 Hard Disc Drive Systems That Convert Your IBM* PC To Perform Like The PC XT!

Maynard Electronics introduces three Winchester Hard Disc Drive Systems — the only drive systems to offer you 10 Mega-bytes of formatted capacity with complete internal installation! These systems offer the user countless benefits and features: capability of booting off the hard disc; additional functions while requiring only **one** card slot in your PC; and, use of available power, thereby preventing overheating problems which have affected other drives. Handling heavyweight data was never easier.

All three systems are quality engineered and work with DOS 2.0 without any special software drivers and also run with other operating systems designed to make use of the XT hard drive system. All you need is the IBM* DOS 2.0 Manual and you're ready to run!

Each system is equipped with a low-power hard disc drive, complete software, cable, a SandStar™ Card and Hard Disc Controller Module. SandStar™ is the first family of modular peripherals created for the IBM* PC. Simple instructions for easy installation are included and all components are backed by an Unconditional One Year Parts and Labor Guarantee.

WS 1

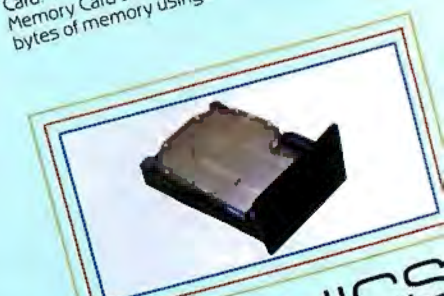
This System is equipped with the SandStar™ Multi-function card. In addition to the Hard Disc Controller Module, you can add up to three other SandStar™ Modules while using only **one** card slot. The following modules are available: Serial Port, Parallel Port, Clock Calendar, Game Adaptor, and Prototyping Module.

WS 2

This System is equipped with the SandStar™ Floppy Drive Controller Card. The Card can control, in addition to the Hard Disc Drive, two floppy drives mounted inside your PC and optionally two additional 5¼" or 8" drives mounted externally. This leaves three system slots for other expansion boards.

WS 3

This System is equipped with the SandStar™ Memory Card. In addition to controlling the Hard Disc Drive, the Memory Card allows you to add 64K bytes to 576K bytes of memory using only one card slot.



To expand your PC to perform like the PC XT, one of our Winchester Hard Disc Drive Systems is right for you. And if you have already made the wise decision to install any of Maynard's SandStar™ Cards, the SandStar™ Hard Disc Controller Module may be purchased separately.

TO ORDER, CONTACT YOUR LOCAL DEALER OR DISTRIBUTOR.

MAYNARD ELECTRONICS
400 East Semoran Blvd.
Casselberry, Florida 32707
305/331-6402

We make modern times better.

NOW! Compatible with COMPAQ!**

*IBM is a trademark of the International Business Machines Corporation.
**COMPAQ is a trademark of the COMPAQ Computer Corporation.

need to know assembler to build or use Getspace, although at the end of the article we'll take a brief look at the assembler source file from which the routine was originally created.

How Does Getspace Do It? Since Getspace operates in all versions of DOS, and since DOS 2.0 handles the FAT (File Allocation Table) in a manner different from the way earlier versions handled it, we need two separate routines within Getspace.

With versions prior to 2.0, Getspace uses DOS function 1BH (27) to retrieve the FAT, scans through the FAT in search of available space, accumulates the amount of space available, and then reports back to the BASIC program.

The procedure is much simpler under DOS 2.0: Function 36H (42), which returns the information for computing the amount of available space, is called; the result of that function call is then passed to BASIC.

Putting the Subroutine Together. The BASIC listing in figure 1 builds the Getspace subroutine. Lines 170 to 270 poke the code into the memory area addressed by string variable Subrt$. This is where Getspace is stored while the subroutine is being built. Lines 280 through 310 perform a checksum to make sure you entered the data statements correctly. If errors are found, line 400 displays a message telling you the line number of the data statement in error and the program stops. Otherwise, line 360 *bsaves* the subroutine under the name Getspace.

And Now for the Test... The listing in figure 2 demonstrates one way Getspace could be used within a BASIC program. Lines 50 through 100 load the subroutine into string Subrt$. The routine in lines 420 through 460 finds the location of string Subrt$. This routine must be executed before each call to Getspace, since the location of Subrt$ could change if BASIC decides to clean house.

Line 140 defines a string variable, Retspace$, that's eight characters long and filled with spaces. This is the field where the available space computed by Getspace will be returned to the calling BASIC program. Lines 190 through 230 retrieve from the user the letter of the drive whose disk is to be checked for available space. Lines 240 through 280 are the actual call to Getspace. The requested drive (Drb$) and the field for the returned amount of space (Retspace$) both are passed to the subroutine via call parameters.

In line 330, after the call, the Retspace$ field is checked for all spaces. In the Getspace subroutine the drive letter requested is checked for validity. If the drive named is invalid, the subroutine ends and the Retspace$ field still contains spaces; line 340 then informs the user that an invalid drive was specified, and the program waits for a response. If Getspace successfully retrieves the amount of available space, line 350 converts the contents of the Retspace$ string to a double-precision number and line 360 displays this figure and associated drive information on the screen.

Run this demonstration program as many times as you like, specifying a different drive each time. The routine should work with hard disks as well as floppies.

Getspace has numerous uses. As the demonstration shows, the space available can be reported to the user exactly, or the number can be rounded off (2,048 bytes can be reported as 2K, for example). You could also specify in your program a minimum acceptable amount of available space and display a warning message if Getspace reports a value below that amount.

Getspace can also be used in conjunction with the Directry subroutine ("BASIC/Assembly Line," January 1984) to create a package that retrieves the filenames on a disk as well as the space available, exactly like the *files* command in BASIC 2.0.

Now that we've seen what Getspace can do, let's look at the assembly routine that makes all this possible.

```
10 '  ***** BUILD GETSPACE SUBROUTINE *****
20 '
30 '  Computes space available on disk / hard disk
40 '
50 '  WRITTEN BY HOWARD GLOSSER
60 '
70 CLS
80 PRINT "Creating GETSPACE subroutine. . ." : PRINT
90 '
100 ' ** THIS SETS UP STRING LOCATION FOR SUBROUTINE
110 '
120 DEF SEG
130 SUBRT$ = STRING$(248,32)
140 SUBLC% = VARPTR(SUBRT$)
150 FREESPC = PEEK(SUBLC% + 1) + PEEK(SUBLC% + 2) * 256
160 LCN = FREESPC
170 '
180 ' ** THIS BUILDS THE SUBROUTINE
190 '
200 LINENO% = 450
210 FOR STMT = 1 TO 31
220  FOR MEM = 1 TO 8
230    READ DT%
240    POKE LCN,DT%
250    CHECKSUM% = CHECKSUM% + DT%
260    LCN = LCN + 1
270  NEXT
280  READ DT%
290  IF CHECKSUM% <> DT% THEN 400
300  LINENO% = LINENO% + 10
310  CHECKSUM% = 0
320 NEXT
330 '
340 ' ** THIS SAVES THE SUBROUTINE
350 '
360 BSAVE "GETSPACE",FREESPC,244
370 PRINT "GETSPACE subroutine created"
380 END
390 '
400 PRINT "ERROR in DATA STATEMENT - Check line " LINENO% : END
410 '
420 ' ** DATA STATEMENTS TO BUILD SUBROUTINE
430 '
440 DATA   85, 139, 236, 139,  94,   6, 139, 127,  965
450 DATA    1, 185,   8,   0, 176,  32, 252, 243,  897
460 DATA  170, 139,  94,   8, 139, 119,   1, 139,  809
470 DATA    4,  36, 223,  60,  65, 125,   3, 233,  749
480 DATA  205,   0,  80, 180,  48, 205,  33, 134,  885
490 DATA  196,  61,   0,   2, 114,   3, 233, 127,  736
500 DATA    0, 180,  25, 205,  33,  90, 128, 242,  903
510 DATA   64, 254, 202,  58, 194, 116,  17,  80,  985
520 DATA   82, 180,  14, 205,  33,  89,  90, 254,  947
530 DATA  193,  58, 200, 118,   3, 233, 159,   0,  964
540 DATA   82, 180,  27, 205,  33,  88,  82, 134,  831
550 DATA  194, 180,  14, 205,  33,  90, 139, 251, 1106
560 DATA  138,   5, 139, 241,  60, 254, 116,   2,  955
570 DATA  209, 230, 131, 199,   3, 139, 202,  51, 1164
580 DATA  192,  51, 210,  80,  82, 139,   5,  71,  830
590 DATA   37, 255,  15,  61,   0,   0, 117,  10,  495
600 DATA  248,  90,  88,   3, 198, 115,   1,  66,  809
610 DATA   80,  82,  73, 227,  28, 139,   5, 131,  765
620 DATA  199,   2,  81, 177,   4, 211, 232,  89,  995
630 DATA   61,   0,   0, 117,  10, 248,  90,  88,  614
640 DATA    3, 198, 115,   1,  66,  80,  82, 226,  771
650 DATA  204, 140, 192, 142, 216, 235,  24, 144, 1297
660 DATA   90, 128, 242,  64, 180,  54, 205,  33,  996
670 DATA   61, 255, 255, 116,  50,  51, 210, 247, 1245
680 DATA  225, 135, 217, 247, 225,  80,  82, 139, 1350
690 DATA   94,   6, 139, 127,   1, 131, 199,   7,  704
700 DATA   90,  88, 190,  10,   0,  80, 139, 194,  791
710 DATA   51, 210, 247, 246,  89,  80, 139, 193, 1255
720 DATA  247, 246,  94, 128, 202,  48, 136,  21, 1122
730 DATA   79, 135, 214,  11, 192, 117, 227,  93, 1068
740 DATA  202,   4,   0, 203,   0,   0,   0,   0,  409
```

Figure 1.

Getspace—More than One Way. The commented assembly listing in figure 3 is self-explanatory. For background on the FAT, see "System Notebook," February 1983, or Appendix C of your DOS manual. For more information about the function calls used in Getspace, see Appendix D of the DOS manual.

The subroutine in lines 9 through 25 retrieves the drive letter that has been requested for the Getspace inquiry. Next, lines 26 through 33 check for the version of DOS being used. As mentioned earlier, the method for computing the available space depends on which version of DOS is running. In versions of DOS before 2.0 (we'll call this the old way for simplicity's sake), a memory image of the FAT was always available for the current default drive. Such is not the case under 2.0 (the new method).

Once the correct version has been established, the appropriate section of code is executed. Lines 34 through 115 are for the old, and lines 116 through 128 are for the new. As you can see by the size of the routines, finding the space under 2.0 is much simpler.

In the old way, we must first determine that the drive requested is valid; this is done in lines 34 through 51. We first find the current default drive and save it (function call 19H). Then we set the requested drive as the new default (function call 0EH); this call returns in register AL a count of the number of drives on the system, and we check this value against the requested drive to see if the drive named is valid.

If the drive isn't valid, the subroutine ends and an error message is displayed. If the drive is valid, the routine continues with a call to function 1BH under INT 21H. This call builds in memory an image of the FAT for the default drive. After the call, the ES:BX pair of registers points to this FAT. Next, the default drive that was saved earlier is restored, so when we exit from Getspace the default drive will be the same as when we entered.

After any necessary adjustments for double-sided disks are made, the AX and DX registers are cleared. These will be used together as a doubleword counter to accumulate the available space. AX is the low word, and DX the high. Lines 75 through 115 do the hefty work in this routine. Here, the program scans the FAT, looking at one-and-a-half-byte intervals for entries of 000, which indicate available space. As these are found, the program adds to AX and DX appropriately. At line 112, the entire FAT has been examined and AX:DX contains the count of available bytes.

Now let's look at how easy it is in DOS 2.0.

One Call, That's All. For DOS 2.0, we execute the routine at lines 116 through 128. A couple of multiplications and a call to function 36H is all we need to compute available space. The function also lets us know (by returning FFFF in AX) if the drive letter is invalid. What could be simpler?

The final step for all versions of DOS is to retrieve from the stack the variable in which the amount of space is to be returned (lines 129 through 134) and to convert the hex value in AX:DX to decimal. This is done in lines 135 through 151, where a double divide operation is repeated as many times as necessary. The double divide prevents an overflow in situations where AX:DX contains a very large number (such as a 10M figure for a hard disk). Finally, we need to restore the BP register and return to Basic, recognizing that there are two parameters on the stack.

In the next Basic/Assembly line, we'll build a routine that will allow you to dump and view files. ▲

```
10 ' ***** THIS PROGRAM DEMONSTRATES GETSPACE *****
20 '
30 ' **** STORE GETSPACE SUBROUTINE IN STRING
40 '
50 KEY OFF
60 DEF SEG
70 SUBRT$ = STRING$(244,32)
80 SUBLC% = VARPTR(SUBRT$)
90 GOSUB 420
100 BLOAD "GETSPACE",GETSPACE
110 '
120 ' ** DEFINE CONSTANTS NECESSARY IN PROGRAM
130 '
140 RETSPACE$ = STRING$(8,32)
150 CLS
160 '
170 ' ** SELECT DRIVE FOR READING DIRECTORY
180 '
190 LOCATE 1,17
200 PRINT "*** DEMONSTRATE GETSPACE SUBROUTINE ***"
210 LOCATE 3,10,1
220 PRINT "Specify drive letter for GETSPACE inquiry : ";
230 GOSUB 480 : DRV$ = KY$ : PRINT DRV$
240 '
250 ' ** CALL TO GETSPACE SUBROUTINE
260 '
270 LOCATE ,,0 : GOSUB 420
280 CALL GETSPACE (DRV$,RETSPACE$)
290 '
300 ' ** LIST RESULTS OF GETSPACE INQUIRY
310 '
320 LOCATE 5,10
330 IF RETSPACE$ <> STRING$(8,32) THEN 350
340 SOUND 50,7 : PRINT "Invalid drive letter for system specified!" : GOTO 380
350 RETSPC# = VAL(RETSPACE$)
360 PRINT "Disk on drive " DRV$ ": has " RETSPC# " Bytes free"
370 BEEP : PRINT
380 PRINT : PRINT "Press SPACE BAR to continue or (S) to Stop "
390 GOSUB 480 : CN$ = KY$
400 IF CN$ = "S" OR CN$ = "s" THEN END
410 GOTO 150
420 '
430 ' ** RETRIEVE LOCATION OF SUBROUTINE
440 '
450 GETSPACE = PEEK(SUBLC% + 1) + PEEK(SUBLC% + 2) * 256
460 RETURN
470 '
480 ' **** KEYIN ROUTINE
490 '
500 KY$ = INKEY$ : IF KY$ = "" THEN 500
510 RETURN
```

Figure 2.

```
1                       ; GETSPACE
2                       ;
3                       ; THIS ROUTINE COMPUTES USABLE SPACE
                          AVAILABLE ON
4                       ; DISKETTE OR HARD DISK AND IS CALLED
                          FROM BASIC
5                       ;
6                       ; WRITTEN BY HOWARD GLOSSER
7                       ;
8  0000                 CSEG                    SEGMENT
9  0000                 GETSPACE                PROC FAR
10                                ASSUME        CS:CSEG
11 0000  55                       PUSH          BP            ;SAVE BP FOR FAR RETURN
12 0001  8B EC                    MOV           BP,SP         ;MOVE STACK POINTER TO BP
13 0003  8B 5E 06                 MOV           BX,[BP]+6     ;POINT BX AT PARM 2
14 0006  8B 7F 01                 MOV           DI,1[BX]      ;GET RETURN FIELD
15 0009  B9 0008                  MOV           CX,8          ;SET UP CX FOR STOSB COUNTER
16 000C  B0 20                    MOV           AL,' '        ;SET UP AL FOR STOSB
17 000E  FC                       CLD                         ;MOVE FORWARD
18 000F  F3/ AA                   REP           STOSB         ;FILL RETURN FIELD WITH BLANKS
19 0011  8B 5E 08                 MOV           BX,[BP]+8     ;POINT BX AT PARM 1
20 0014  8B 77 01                 MOV           SI,1[BX]      ;GET DRIVE LETTER
21 0017  8B 04                    MOV           AX,[SI]       ;MOVE DRIVE LETTER TO AX
22 0019  24 DF                    AND           AL,0DFH       ;MAKE SURE IT'S CAPS
23 001B  3C 41                    CMP           AL,41H        ;IS LETTER LESS THAN A?
24 001D  7D 03                    JGE           CKVER         ;NO - CONTINUE
25 001F  E9 00EF R                JMP           EXITSPC       ;YES - INVALID/GET OUT
26 0022                 CKVER:
27 0022  50                       PUSH          AX            ;SAVE DRIVE LETTER
28 0023  B4 30                    MOV           AH,30H        ;SET AH FOR FUNCTION CALL 30H
29 0025  CD 21                    INT           21H           ;DOS INTERRUPT
30 0027  86 C4                    XCHG          AL,AH         ;EXCHANGE AL AND AH
31 0029  3D 0200                  CMP           AX,0200H      ;VERSION 2.0?
32 002C  72 03                    JB            SYSDRV        ;NO - GO CHECK DRIVES ON SYSTEM
33 002E  E9 00B0 R                JMP           SPACE20       ;YES - GO TO 2.0 ROUTINE
34 0031                 SYSDRV:
35 0031  B4 19                    MOV           AH,19H        ;SET UP TO FIND DEFAULT DRIVE
36 0033  CD 21                    INT           21H           ;DOS INTERRUPT
37 0035  5A                       POP           DX            ;RESTORE DRIVE LETTER
38 0036  80 F2 40                 XOR           DL,40H        ;CONVERT DRIVE LETTER TO NUMBER
39 0039  FE CA                    DEC           DL            ;ADJUST DRIVE NUMBER
40 003B  3A C2                    CMP           AL,DL         ;IS REQUESTED DRIVE THE DEFAULT?
41 003D  74 11                    JE            GETFAT        ;YES - GO GET FAT
42 003F  50                       PUSH          AX            ;NO - SAVE DEFAULT
43 0040  52                       PUSH          DX            ;NO - SAVE REQUESTED DRIVE
44 0041  B4 0E                    MOV           AH,0EH        ;SET NEW DEFAULT
45 0043  CD 21                    INT           21H           ;DOS INTERRUPT
46 0045  59                       POP           CX            ;RESTORE REQUESTED DRIVE
47 0046  5A                       POP           DX            ;RESTORE DEFAULT
```

```
0047  FE C1                  INC    CL              ;ADJUST FOR TRUE DRIVE NUMBER
0049  3A C8                  CMP    CL,AL           ;IS REQUESTED DRIVE VALID?
004B  76 03                  JBE    GETFAT          ;YES - GET FAT
004D  E9 00EF R              JMP    EXITSPC         ;NO - GET OUT
0050              GETFAT:
0050  52                     PUSH   DX              ;SAVE OLD DEFAULT
0051  B4 1B                  MOV    AH,1BH          ;SET UP AH FOR FAT CALL
0053  CD 21                  INT    21H             ;DOS INTERRUPT
0055  58                     POP    AX              ;RESTORE OLD DEFAULT DRIVE
0056  52                     PUSH   DX              ;SAVE FAT
0057  86 C2                  XCHG   AL,DL           ;EXCHANGE DL AND AL
0059  B4 0E                  MOV    AH,0EH          ;SET DEFAULT BACK
005B  CD 21                  INT    21H             ;DOS INTERRUPT
005D  5A                     POP    DX              ;RESTORE FAT
005E  8B FB                  MOV    DI,BX           ;POINT DI AT FAT
0060  8A 05                  MOV    AL,BYTE PTR [DI] ;MOVE FIRST BYTE OF FAT TO AL
0062  8B F1                  MOV    SI,CX           ;MOVE SECTOR SIZE TO SI
0064  3C FE                  CMP    AL,0FEH         ;SINGLE-SIDED DISK?
0066  74 02                  JE     SNGLSIDE        ;YES - SKIP THE SHIFT BELOW
0068  D1 E6                  SHL    SI,1            ;NO - TWO SECTORS PER CLUSTER
006A              SNGLSIDE:
006A  83 C7 03               ADD    DI,3            ;POINT PAST FIRST 2 ENTRIES
006D  8B CA                  MOV    CX,DX           ;CX CONTAINS # OF CLUSTERS
006F  33 C0                  XOR    AX,AX           ;CLEAR AX/SPACE AVAILABLE LOW WORD
0071  33 D2                  XOR    DX,DX           ;CLEAR DX/SPACE AVAILABLE HIGH
                                                     WORD
0073  50                     PUSH   AX              ;SAVE AX
0074  52                     PUSH   DX              ;SAVE DX
0075              SCANFAT:
0075  8B 05                  MOV    AX,[DI]         ;MOVE WORD TO AX
0077  47                     INC    DI              ;BUMP DI
0078  25 0FFF                AND    AX,0FFFH        ;STRIP OFF HIGH BITS
007B  3D 0000                CMP    AX,00H          ;IS IT ZERO (AVAILABLE)?
007E  75 0A                  JNE    NEXTENT         ;NO - CHECK NEXT FAT ENTRY
0080  F8                     CLC                    ;CLEAR CARRY FLAG
0081  5A                     POP    DX              ;RESTORE DX
0082  58                     POP    AX              ;RESTORE AX
0083  03 C6                  ADD    AX,SI           ;ADD SECTOR SIZE TO AVAIL COUNT
0085  73 01                  JNC    STRSPC1         ;IS THERE A CARRY?
0087  42                     INC    DX              ;YES, ADD TO HIGH WORD
0088              STRSPC1:
0088  50                     PUSH   AX              ;SAVE AX
0089  52                     PUSH   DX              ;SAVE DX
008A              NEXTENT:
008A  49                     DEC    CX              ;DECREASE FAT ENTRY BY 1
008B  E3 1C                  JCXZ   END1            ;IF CX = 0 THEN END
008D  8B 05                  MOV    AX,[DI]         ;MOVE WORD TO AX
008F  83 C7 02               ADD    DI,2            ;MOVE TO NEXT WORD
0092  51                     PUSH   CX              ;SAVE FAT ENTRY COUNT
0093  B1 04                  MOV    CL,4            ;SET UP CL FOR SHIFT
0095  D3 E8                  SHR    AX,CL           ;SHIFT CONTENTS RIGHT
0097  59                     POP    CX              ;RESTORE CX
0098  3D 0000                CMP    AX,00H          ;IS IT ZERO (AVAILABLE)?
009B  75 0A                  JNE    NOTAVAIL        ;NOPE - DO NEXT FAT ENTRY
009D  F8                     CLC                    ;CLEAR CARRY FLAG
009E  5A                     POP    DX              ;RESTORE SPACE AVAILABLE (HIGH)
009F  58                     POP    AX              ;RESTORE SPACE AVAILABLE (LOW)
00A0  03 C6                  ADD    AX,SI           ;ADD SECTOR SIZE TO AVAIL COUNT
00A2  73 01                  JNC    STRSPC2         ;IS THERE A CARRY?
00A4  42                     INC    DX              ;YES, ADD TO HIGH WORD
00A5              STRSPC2:
00A5  50                     PUSH   AX              ;SAVE AX
00A6  52                     PUSH   DX              ;SAVE DX
00A7              NOTAVAIL:
00A7  E2 CC                  LOOP   SCANFAT         ;CONTINUE SCANNING FAT
00A9              END1:
00A9  8C C0                  MOV    AX,ES           ;MOVE ES TO AX
00AB  8E D8                  MOV    DS,AX           ;MOVE AX TO DS
00AD  EB 18 90               JMP    ENDSPC          ;GO TO END
00B0              SPACE20:
00B0  5A                     POP    DX              ;RESTORE DRIVE LETTER
00B1  80 F2 40               XOR    DL,40H          ;CONVERT LETTER TO NUMBER
00B4  B4 36                  MOV    AH,36H          ;SET UP FOR FUNCTION CALL 36H
00B6  CD 21                  INT    21H             ;DOS INTERRUPT
00B8  3D FFFF                CMP    AX,0FFFFH       ;INVALID DRIVE?
00BB  74 32                  JE     EXITSPC         ;YES - INVALID/GET OUT
00BD  33 D2                  XOR    DX,DX           ;CLEAR DX
00BF  F7 E1                  MUL    CX              ;MULTIPLY AX BY CX
00C1  87 D9                  XCHG   BX,CX           ;EXCHANGE BX AND CX
00C3  F7 E1                  MUL    CX              ;MULTIPLY AX BY CX
00C5  50                     PUSH   AX              ;SAVE AX
00C6  52                     PUSH   DX              ;SAVE DX
00C7              ENDSPC:
00C7  8B 5E 06               MOV    BX,[BP]+6       ;POINT BX AT PARM 2
00CA  8B 7F 01               MOV    DI,1[BX]        ;GET RETURN FIELD
00CD  83 C7 07               ADD    DI,7            ;POINT TO END OF RETURN FIELD
00D0  5A                     POP    DX              ;RESTORE SPACE AVAILABLE (HIGH)
00D1  58                     POP    AX              ;RESTORE SPACE AVAILABLE (LOW)
00D2              HEXTODEC:
00D2  BE 000A                MOV    SI,10           ;SET UP DIVISOR FOR CONVERSION
00D5  50                     PUSH   AX              ;SAVE SPACE AVAILABLE (LOW)
00D6  8B C2                  MOV    AX,DX           ;MOVE SPACE AVAILABLE (HIGH) TO AX
00D8  33 D2                  XOR    DX,DX           ;CLEAR DX
00DA  F7 F6                  DIV    SI              ;DIVIDE AX BY SI - DIVIDE #1
00DC  59                     POP    CX              ;RESTORE SPACE AVAILABLE (LOW)
00DD  50                     PUSH   AX              ;SAVE QUOTIENT OF FIRST DIVIDE
00DE  8B C1                  MOV    AX,CX           ;MOVE SPACE AVAILABLE (LOW) TO AX
00E0  F7 F6                  DIV    SI              ;DIVIDE AX BY SI - DIVIDE #2
00E2  5E                     POP    SI              ;RESTORE QUOTIENT OF FIRST DIVIDE
00E3  80 CA 30               OR     DL,30H          ;CHANGE DL TO DECIMAL DIGIT
00E6  88 15                  MOV    BYTE PTR [DI],DL ;MOVE DECIMAL FIGURE TO [DI]
00E8  4F                     DEC    DI              ;BACK UP 1 POSITION IN RETURN FIELD
00E9  87 D6                  XCHG   DX,SI           ;QUOTIENT OF DIVIDE #1 GOES IN DX
00EB  0B C0                  OR     AX,AX           ;ARE WE DONE?
00ED  75 E3                  JNZ    HEXTODEC        ;NOT ZERO - DIVIDE AGAIN
00EF              EXITSPC:
00EF  5D                     POP    BP              ;RESTORE BP
00F0  CA 0004                RET    4               ;FAR RETURN WITH 2 PARMS
00F3  CB                     RET
00F4              GETSPACE          ENDP
00F4              CSEG              ENDS
                  END
```

Figure 3.

Basically Speaking

by John Dickinson

Math, Modes, *For* and *Next*

You're not ready to program your own version of *1-2-3* or *Pac-Man* yet, but after just one month you have learned some of the important basics of Basic, and you can congratulate yourself for coming this far so fast.

Last month you learned how to start Basic on your PC or Junior and about Basic statements that compute mathematical formulas and display results. You saw that Basic looks a lot like the algebra you were taught in school, and therefore it seemed pretty familiar to you. You even began to use a little of a computer's advantage over a calculator when you learned how easily Basic could repeat calculations using different values for variables.

This month we're going to see a few more similarities between Basic and algebra—and more of the differences you need to be aware of as well. We'll also learn how to make permanent copies of Basic programs (so you can use them whenever you want without having to reenter them) and an easier way to make Basic repeat itself.

We'll continue using the temperature conversion program from last month, so let's start by reviewing it. If you haven't already started up Basic, insert the disk you made last month and type

basic

to get things going, or just turn on your machine if you're using Cassette Basic or Cartridge Basic without PC-DOS.

The Basic program we used to convert Celsius temperatures to Fahrenheit looked like this (you may want to type it in again):

```
let temp.celsius = 10
let temp.fahrenheit = 32 + temp.celsius * 9 / 5
```

A program is made up of a series of Basic statements much like these. Statements that begin with *let* are called *assignment statements* and bear a close resemblance to algebraic equations.

The first statement in our program assigns a Celsius temperature of 10 to a Basic variable we have named *temp.celsius*; the second converts *temp.celsius* to its Fahrenheit equivalent, assigning the result to a variable named *temp.fahrenheit*.

After typing this much in and pressing the Enter key on each line to make Basic execute the statements, we learned how to display the values of both variables using a Basic output statement, like this:

```
? temp.fahrenheit, temp.celsius
50  10
```

We also added the following statement to increase the value of *temp.celsius*:

```
let temp.celsius = temp.celsius + 10
```

and used the cursor keys to make Basic repeat the computation of *temp.fahrenheit* using the new value for *temp.celsius*. We did this by moving the cursor to the statement on the screen that we wanted Basic to execute again and then pressing the Enter key to tell Basic to go ahead and execute it. We saw that it's easy to make Basic keep on adding 10 to *temp.celsius* and to get it to repeat the computation of *temp.fahrenheit* several times. We'll come back to repetition later on this month when we learn an easier way of getting Basic to repeat a calculation like this one.

It was suggested that you try to turn the program around and have Basic convert Fahrenheit temperatures to the Celsius scale. Did you try it? Did it work? Let's review the process of changing the program around because it involves the correct use of both mathematics and Basic, and because it's important to make sure you've got it all right.

In order to develop a statement for converting Fahrenheit temperatures to Celsius, we have to invert our original equation mathematically. Inverting an algebraic equation such as this one is a logical process requiring that each arithmetic operation be inverted individually; when that's been done, the entire formula will have been inverted. In our case the inversion will convert the formula so that it computes Celsius equivalents from Fahrenheit temperatures.

We must be careful, however, to apply the rules of mathematics correctly or we won't get the right answers. So, let's look at each step involved in inverting the temperature formula algebraically:

1. Original algebraic equation
   let temp.fahrenheit = 32 + temp.celsius * 9 / 5
2. Reverse addition of constant
   let temp.fahrenheit − 32 = temp.celsius * 9 / 5
3. Invert division by constant
   let 5 * (temp.fahrenheit − 32) = temp.celsius * 9
4. Invert multiplication by constant
   let 5 / 9 * (temp.fahrenheit − 32) = temp.celsius
5. Reverse final algebraic equation
   let temp.celsius = 5 / 9 * (temp.fahrenheit − 32)

Each step of the inversion maintains the mathematical relationship between the variables *temp.celsius* and *temp.fahrenheit*, and the equation in step 5 correctly converts a Celsius temperature to Fahrenheit.

Let's test-drive the new formula in Basic to make sure it works. If you have just started Basic, all variables will have been set to 0 automatically, but if you've been using Basic, you should reset any previously defined variables to 0 by typing:

clear

which clears out Basic's memory. You may also want to clear off Basic's screen by typing:

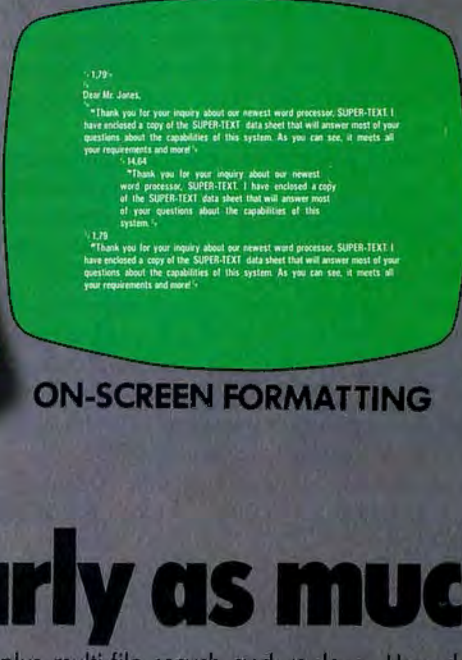
cls

If our new formula is correct, it should convert a Fahrenheit temperature of fifty degrees to ten degrees Celsius (the reverse of what we did earlier). Enter the following to see if it does (be sure to spell everything correctly and type things in just as you see them):

Come in late on "Basically Speaking"? All back issues of the column—from March 1984— are still available; for further information, see page 4.

SUPER-TEXT Professional does everything the competition does, except one thing.

ON-LINE "HELP"



ON-SCREEN FORMATTING

It doesn't cost nearly as much.

SUPER-TEXT PROFESSIONAL FOR IBM* provides the most powerful word processing features on the market today at a sensible price. Super-Text makes word processing easy. A few of its many features are valuable **On-Screen Formatting**, easy to use **On-Screen Help Reference Guides**, professional **Split-Screen capability**, and **Autolink**.

Super-Text Professional utilizes all of the IBM function keys. It provides automatic page headers and footers, automatic page and chapter numbering, and the Preview Mode allows you to check page endings.

Super-Text provides for easy text editing. Special block operations for text copy, save and delete. Automatic tabbing and formatting plus multi-file search and replace. User definable characters allow you to make full use of most printer functions, including multi-color printing, superscripting and subscripting. Plus Super-Text provides full compatibility with IBM Edlin Files.

This very powerful and easy to use word processor can be yours for only **$175.00**.

*Super-Text requires IBM DOS Version 1.1., minimum 64K. Also available for Apple II+ and Apple IIe.

Super-Text Professional.
Also Available For Apple IIe $175, COMMODORE 64 $99, ATARI 800, 1200 $99.

MUSE®
SOFTWARE

347 North Charles Street, Baltimore, Maryland 21201 301/659-7212

```
let temp.fahrenheit = 50
let temp.celsius = 5 / 9 * (temp.fahrenheit - 32)
? temp.fahrenheit, temp.celsius
50  10
```

Good, we got it right! But last month you were warned that you might run into trouble trying to do this conversion on your own. It looks so simple, you might be wondering what the catch could be (unless you already got caught!).

There are actually two things that can go wrong on you. Both were carefully taken care of above when we did the algebraic conversion, but it's there that you're most likely to get into trouble. The first problem has to do with the use of parentheses in mathematics and BASIC, and the second has to do with a major difference between algebraic equations and BASIC assignment statements. We'll take up the parentheses first.

Parentheses are used in algebra to hold the result of a mathematical operation aside so that additional operations can be applied to the result instead of being treated as a component of the first operation. Parentheses may be used anywhere, but they are normally used to override the rules of mathematical precedence or priority.

What does that mean? In any arithmetic expression several mathematical operators may be present, and the rules of mathematical priority dictate which operations must be performed first, second, and so on. The normal mathematical rules of precedence are:

1. Exponentiation is performed first.
2. Multiplication and division are performed next.
3. Addition and subtraction are performed last.

For example, the expression:

```
5 * 50 - 32
```

requires the following sequence of calculations to be made:

```
5 * 50 = 250 (multiplication is done first)
250 - 32 = 218 (followed by subtraction)
```

The rules of mathematical precedence require that we multiply 50 by 5 before subtracting 32. If, instead, we want to multiply the difference between 50 and 32 by 5, we can override the rules of precedence by using parentheses, like this:

```
5 * (50 - 32)
```

In this case the correct sequence of calculations becomes:

```
(50 - 32) = 18 (Operation in parens is done first)
5 * 18  = 90 (followed by multiplication)
```

which gives us a different result altogether.

BASIC uses the same rules of mathematical precedence and uses parentheses to override them in the same way. To prove it to yourself, type the following arithmetic expressions into BASIC:

```
? 5 * 50 - 32
 218
? 5 * (50 - 32)
 90
```

You can use parentheses anywhere you think they may be needed. And if you use them where they aren't needed, well, no harm is done. For example:

```
? 5 / 9 * (50 - 32)
 10
```

gives the same result as:

```
? (5 / 9) * (50 - 32)
 10
```

and:

```
? 5 * (50 - 32) / 9
 10
```

because division and multiplication have the same priority. On the other hand,

```
? (5 / 9) * 50 - 32
-4.222222
```

gives a different answer because, without the parentheses, multiplication and division are done before either addition or subtraction.

The last couple of examples should look somewhat familiar to you, since they're derived from our temperature conversion program. In fact, incorrect use of mathematical priority would probably lead you to put this statement:

```
let temp.celsius = 5 / 9 * temp.fahrenheit - 32
```

in the program. If you substitute this statement for the correct one, you'll get the same incorrect answer (−4.222222).

There are a few other rules of mathematical precedence used by BASIC, but they involve operators we haven't come to yet. We'll revise BASIC's rules of precedence when we get to the other operators.

The other problem you may have run into is a little simpler to deal with, but it's the result of a mistake people often make when first learning to program. If you look back at the steps required for inverting the algebraic formula, you'll notice that we added a step at the end that is mathematically unnecessary. In algebra it is perfectly okay to state:

```
let 5 / 9 * (temp.fahrenheit - 32) = temp.celsius
```

But in BASIC it's not. If you type this statement into BASIC, the message

```
Syntax error
```

will be displayed on your screen.

Why is this an error in BASIC? It's an error because a BASIC assignment statement is similar to an algebraic equation, but it's not the same! An algebraic equation states a mathematical relationship that will hold regardless of which side of the equals sign ("=") variables and constants appear on. A BASIC assignment statement, on the other hand, calculates a mathematical formula and assigns the results to a variable on the left side of the BASIC assignment operator (which is also "="). Only one variable at a time can be assigned in a BASIC statement and, as

a result, only one variable may appear on the left side of a BASIC assignment operator.

Another BASIC rule is that a constant may never appear on the left side of an assignment statement. You may want to prove this to yourself by typing:

```
let 20 = temp.fahrenheit
Syntax error
```

It might be a good idea to try doing the temperature conversion using incorrect formulations of the assignment statement, just to see the different results you can get. The following examples should prove interesting:

```
let temp.celsius = 5 / 9 * temp.fahrenheit - 32
let temp.celsius + 32 = 5 / 9 * temp.fahrenheit
let temp.fahrenheit = 9 * temp.celsius + 32 / 5
let temp.fahrenheit = temp.celsius * 9 / 5 + 32
let temp.celsius / 5 = 9 * temp.fahrenheit - 32
```

In each case, go ahead and display the results to see what happens. You may also find it handy to do the repetition exercise from last month, this time converting a series of Fahrenheit temperatures to their Celsius equivalents.

So far, all of our programming has been done using BASIC's immediate mode. This has allowed us to try things in BASIC and get results without waiting, but it hasn't allowed us to save programs. Whenever we decided to run a program we'd used before, we had to enter the entire sequence of BASIC statements all over again.

Not surprisingly, there is a way to enter BASIC programs and subsequently save them; it's BASIC's *deferred mode*. In this mode, BASIC doesn't act on each statement when you press the Enter key. Instead, statements are saved in your PC's memory until you want to run, or *execute*, the whole program you've entered. The program that's saved in memory can also be saved on disk or tape for use at a later time. Execution is deferred until you tell BASIC to run the program—which tells you where deferred mode got its name.

The deferred mode in BASIC is not as quick to give you answers as immediate mode (since it waits for you to tell it when to act), but it's still a very quick way to enter and use programs. Don't forget about immediate mode, however. As you'll see shortly, immediate mode comes in handy when you're trying to debug programs to find out what's wrong (or right) with them, and it will always be useful for doing quick calculations and for simply trying out various formulations of a statement to see how they work (as we did earlier).

While learning about BASIC's deferred mode, you'll also be learning some other things. There are things to learn about BASIC's screen and editor, a new rule about naming BASIC variables and commands, and a new use for the Enter key. Something else you'll learn about is *line numbering*. Line numbers are very important in BASIC's deferred mode, so we'll start changing the original temperature conversion program by adding line numbers to it.

When we did the temperature conversion in immediate mode, the first statement in the program was:

```
let temp.celsius = 10
```

All we have to do to use this statement in BASIC's deferred mode is give it a line number. Why a line number? In deferred mode BASIC saves each statement in memory, and a statement's line number tells BASIC where to save it relative to other statements in the program. The line numbers in a program also determine the order in which BASIC will execute statements in deferred mode.

Clear out BASIC's memory and screen (type *clear* and then *cls*); then give our assignment statement a line number by typing:

```
10 let temp.celsius = 10
```

Press the Enter key as you did before. The Enter key is still the "go" button, but when used from deferred mode it tells BASIC to save the statement in memory rather than execute it immediately. You may notice that BASIC's Ok did not appear this time after you pressed the Enter key; its nonappearance indicates that no statements were executed. To prove to yourself that *temp.celsius* hasn't yet been assigned the value of 10, type:

```
? temp.celsius
0
```

This time the Ok appears, indicating that the display statement (the ?) has been executed. The assignment statement, however, hasn't been executed yet, so *temp.celsius* still has a value of 0. All that happened when you entered the first statement was that BASIC saved the statement as "line 10."

Line 10 is now stored in your computer's memory and you can list it on your screen by typing:

```
list
10 LET TEMP.CELSIUS = 10
```

Hmmm . . . what's with the uppercase letters? BASIC has an additional rule for naming variables and commands that we didn't mention before. Alphabetic characters must always be uppercase. You don't usually have to worry about this rule because BASIC automatically translates letters to uppercase when you enter lowercase ones. When you use immediate mode, all the letters you type are actually translated internally by BASIC (unless you type uppercase letters in the first place), and that's the reason we didn't bother to mention the rule before now.

In these articles, we'll continue using lowercase letters to indicate what should be typed in at the keyboard. And you can continue typing them in whatever case you prefer. When BASIC commands and variables appear in the text of these articles we'll italicize them, to indicate that we're talking about a BASIC command.

At this point you have a one-line program that assigns a value to *temp.celsius*. That's not much, but it is a deferred-mode BASIC program, so let's run it to see if it works. How do you do that? Type:

```
run
```

and BASIC's friendly Ok will quickly appear. In deferred mode, the Ok prompt means that the saved program has been executed. Has the program done anything? Let's use immediate mode to find out. Type:

```
? temp.celsius
10
```

Good, we got *temp.celsius* assigned the value 10, just as we wanted. But—is our program still there?

```
list
10 LET TEMP.CELSIUS = 10
```

So far, so good; the program is saved in memory and runs correctly. Now let's see if we can get the rest of our program to work in deferred mode.

Since line 10 is already in memory, you don't have to retype it. All you have to do is add the rest of your temperature conversion program. When we entered the program in immediate mode, the rest of the statements came after the first assignment statement (the one you just entered), and we should arrange the program in the same way in deferred mode. How do we do that? Just use successively larger line numbers for the next two statements of the program, like this:

```
20 let temp.fahrenheit = 32 + temp.celsius * 9 / 5
30 ? temp.fahrenheit, temp.celsius
```

Now, list the whole program to make sure everything's in good order:

```
list
10 LET TEMP.CELSIUS = 10
20 LET TEMP.FAHRENHEIT = 32 + TEMP.CELSIUS * 9 / 5
30 PRINT TEMP.FAHRENHEIT, TEMP.CELSIUS
```

Oops. What happened to the ? we typed on line 30? When we typed the program in originally, we entered a question mark on line 30 because we wanted to ask BASIC to display the value of our variables on the video screen. Now we see the word PRINT instead!

The ? is BASIC's abbreviation for PRINT, and all along we've been using the abbreviation instead of trying to explain why BASIC's video display command is PRINT. (Okay, we're lazy.)

Why is BASIC's display command named PRINT? Well, when BASIC was invented, the only type of computer display generally available was a printer, much like the one you probably have attached to your machine. As a result, it seemed natural to BASIC's inventors to use the word PRINT to tell BASIC to display results, and the name of the command has stuck throughout BASIC's life. There are other anomalies in BASIC, many of which have their roots in the history of the language. We'll point out the more interesting ones as we go along.

Back to business. If you type *run* now, you'll see:

```
run
50  10
```

displayed on your screen. Not surprisingly, these are the same results we got when we ran this program with the same value (10) for *temp.celsius* in immediate mode. You can also type *list* again and see the program listed on the screen or use immediate mode to display the values of any of the variables.

You'll be making frequent use of both *list* and *run*, so let's get acquainted with an easier way to use both of them. If you look along the bottom of BASIC's screen you'll notice several (ten on an eighty-character screen and five on a forty-character screen) short words displayed, each preceded by a number. Each of the displayed words is a BASIC command, and the number preceding it tells you which function key on your PC can be used to issue the command.

Pressing the F1 key, followed by the Enter key, issues the *list* command to BASIC. Notice that it says *1list* at the lower left of your screen, indicating this use of the F1 key. To list your program using the F1 key,

type

[F1][Enter]

The word

LIST

will appear on your screen, followed by your program.

Next to the *1list* at the bottom of your screen you'll see *2run*<-. That tells you that pressing the F2 key causes BASIC to *run* your program; the arrow to the right of *2run* means that you don't have to press the Enter key; just press F2, and BASIC will display *run* and execute your program just as if you had typed the *run* command (and pressed Enter) yourself.

We'll be learning about a few of the more useful function keys this month, and later we'll see how to change the meanings of the function keys so they'll do anything we want them to. Meanwhile, give these first two a try to see how they work.

Even though you've learned how to use BASIC's deferred mode, you still are faced with the prospect of losing your program if you leave BASIC or turn your machine off. You can avoid this by saving the program in a file on disk (or on cassette tape if you don't have a disk drive). Like any other file, a BASIC program needs to have a name, and the name must follow the rules for PC-DOS files (even if you're using cassette tape).

The first step in saving your program, then, is to choose a name for it, and the second step is to tell BASIC to save the program in a file under that name. Let's call our program CONVERTC to indicate that it converts Celsius (C) temperatures to Fahrenheit. Notice that, in compliance with PC-DOS rules, the name is only eight characters in length (it can contain fewer characters if you wish). To *save* the program, type:

save"convertc"

and hit Enter. The quotation marks have a special meaning that we'll discuss later on. Meanwhile, just think of them as how you tell BASIC where the file name begins and ends.

Disobeying the PC-DOS rules for naming files can have various results. If the name you use for saving a BASIC program is more than eight characters long, it will be truncated to the first eight characters you typed. If you use an invalid character you'll get an error message. For example:

save"convert*"
Bad file name

or

save"convert¦"
Too many files

Sharp-eyed readers will probably notice that the F4 key is defined on their screens as *4save* and can be used to *save* files. To *save* the program you've created, just type:

[F4]convertc"

This will appear on your screen as:

SAVE"convertc"

The first quotation mark is inserted by the F4 key, so you don't have to type it.

Let's leave BASIC for a moment so you can see what's happened on your disk. (If you're using BASIC without PC-DOS you can ignore this paragraph, but you might want to come back to it later.) Remember, to leave BASIC, type:

system

and the familiar PC-DOS prompt will return. Issue a directory command (*dir*) and, if you haven't added anything new to the disk you prepared last month, you should see this on your screen:

```
A>dir

Volume in drive A has no label
Directory of A:\
COMMAND COM     17664    3-08-83   12:00p
BASIC COM       16256    3-08-83   12:00p
CONVERTC BAS    107      1-23-84    3:54p
2 File(s) 304128 bytes free
```

Where did CONVERTC get its ".BAS" filename extension? BASIC added it for you, and it's there to let you know which files are your BASIC programs whenever you check your directory. You can add it yourself by typing:

save"convertc.bas"

if you want, but it's easier just to let BASIC do it for you. You can also discover what files are on your disk while you are in BASIC. Return to BASIC and type:

```
files
A:\
COMMAND .COM     BASIC .COM     CONVERTC.BAS
304128 Bytes free
```

which is similar to the display format you get when using the /W (wide display) option of the PC-DOS directory command. The *files* command isn't available if you are not using BASIC with PC-DOS.

Now, can you load the CONVERTC.BAS program back into BASIC and run it again? Certainly! Just type:

load"convertc"

You don't need to type ".BAS". BASIC assumes you want to *load* a BASIC program. After the program is loaded, press the F2 key to run it. You should see the same results as before. Once again, there's a handy function key (it's F3) for you to use, and you can LOAD"CONVERTC.BAS by typing:

[F3]convertc"

which appears on your screen as:

LOAD"convertc"

instead of typing the whole thing in yourself.

Every time you've run your deferred-mode version of CON-

VERTC.BAS it's used a Celsius temperature of 10, which is assigned in line 10 of the program. How can you change this value? There are many ways to do it, but we'll start with the simplest ones and work up from there.

The easiest way to change the Celsius temperature is to reenter line 10 of the program, using a different value for *temp.celsius*. Just type:

```
10 let temp.celsius = 20
```

and run the program (use the F2 key) again. This time you'll see:

```
68   20
```

on your screen. Every time you retype a line number with a new statement, the old statement in memory is replaced.

List the program, and look at line 10 with the new assignment for *temp.celsius* to see for yourself.

Retyping a line number and statement is also an easy way to correct a mistake you've made in a statement. For example, if you entered:

```
20 LET TEMP.FAHRENHEIT = 23 + TEMP.CELSIUS * 9 / 5
```

or made any other mistake in line 20, you could correct it by typing:

```
20 let temp.fahrenheit = 32 + temp.celsius * 9 / 5
```

Typing a line number without a statement deletes the corresponding line from the program. For example, type

```
10
```

hit Enter, and then list the program. Line 10 will have disappeared.

If you run the program now (without line 10) you will see:

```
32   0
```

on your screen. Why 0 for *temp.celsius*? Before it runs a deferred-mode program, BASIC issues a *clear* command program, assuring you that all variables will be set to 0 when you start any BASIC program you write. BASIC's *clear* command set *temp.celsius* to 0 and, since line 10 is no longer present in the program, nothing changed the value of that variable. If you check it out, you'll find that the correct value for *temp.fahrenheit* is 32 when *temp.celsius* has a value of 0.

To refresh your program, just LOAD"CONVERTC" in again from disk (use F3). You might want to list and run the program just to see that all is well and that line 10 has returned.

We've seen how to change the value of *temp.celsius* and even how to restore it to the original value we had when the program was saved on disk, but how can BASIC repeat itself in a deferred-mode program? Like so many other things in BASIC, there are several ways to have a program repeat itself and use a different value for a variable on each repetition, but some methods are preferred over others. What we'll look at next is the BASIC FOR loop, which is about the most common and powerful way to make BASIC repeat a calculation.

What is a FOR loop and how does it work? In the English language, a loop is defined as a length of cord with its ends tied together. If you lay a loop of cord on a table and run your finger along it, you'll find your finger going in a circle—or at least making some kind of continuous elliptic shape.

The thing that distinguishes a loop from any other kind of oval shape is the knot that ties the ends of the cord together. Every time your finger moves over the knot you feel a bump and, at the same time, you find your finger starting at the beginning of the cord once more.

A loop in the BASIC language isn't very different. Instead of being made of a piece of cord, a BASIC FOR loop consists of a group of BASIC statements tied together with a "knot" made up of two new kinds of BASIC statements. The group starts with a FOR statement and ends with a NEXT statement. The FOR statement marks the beginning of the FOR loop, or group of statements, and the NEXT statement marks the end of the loop.

BASIC starts executing a FOR loop when it sees a FOR statement. When BASIC sees a NEXT statement, it bumps and returns to the beginning of the loop (group of statements), marked by the FOR statement.

When BASIC bumps, it does something else, too. It changes the value of something called the *counter*. You initialize the loop counter by assigning it some value in the FOR statement, and its value changes every time the NEXT statement is encountered. And, whereas you might continue running your finger over the cord until you fall asleep, BASIC will only continue bumping through a loop until the loop counter exceeds a target value, which is also set by the FOR statement.

The best part of all this is that *you* get to define what BASIC statements are grouped into a FOR loop, where the loop begins and ends, and what the starting and target values are for the loop counter. Does all this mean that we can use a FOR loop to make our program repeat itself using different values for *temp.celsius*? You bet it does; let's see how.

When we got BASIC to repeat our program last month we had it compute *temp.fahrenheit* FOR values of *temp.celsius* from 10 to 100. BASIC's version of stating the problem in a FOR loop isn't very different from an English-language version of the same thing. Replace line 10 in CONVERTC by this one:

```
10 for temp.celsius = 10 to 100
```

That wasn't so hard. Now, add a new line:

```
40 next temp.celsius
```

The FOR and NEXT pair of statements define a FOR loop that will do what we want.

- The FOR statement tells BASIC to use *temp.celsius* as a loop counter and assigns it an initial value of 10. It also tells BASIC to repeat the FOR loop until the value of *temp.celsius* has exceeded 100.
- The NEXT statement marks the end of the loop and tells BASIC to increase *temp.celsius* by 1. BASIC will always increase a loop counter by 1 unless we tell it to do something else.

Now *list* the newest version of CONVERTC to check it. It should look like this:

```
10 FOR TEMP.CELSIUS = 10 TO 100
20 LET TEMP.FAHRENHEIT = 32 + TEMP.CELSIUS * 9 / 5
30 PRINT TEMP.FAHRENHEIT, TEMP.CELSIUS
```

```
40 NEXT TEMP.CELSIUS
```

Each time it repeats the FOR loop, BASIC repeats execution of line 20 and line 30. It recomputes *temp.fahrenheit* for each value of *temp.celsius* from 10 to 100 (line 20) and displays the results for each one (line 30).

Run the program (use the F2 key), and you'll see a whole bunch of values for *temp.fahrenheit* and *temp.celsius* go flying by on your screen as BASIC repeats the FOR loop. If all is correct, the last values displayed will be:

```
212  100
```

just as they were when we ran the program (doing the repetition with the cursor keys) in immediate mode.

Oh heck. We forgot one thing. Last month we decided that we only needed to convert *temp.celsius* to *temp.fahrenheit* in STEPs of ten degrees. There's another item we can add to the FOR statement that will take care of that. We might as well learn another BASIC editing trick at the same time. Type the following into BASIC:

```
edit 10
```

and line 10 will appear on your screen with the cursor under the first character, like this:

```
10 FOR TEMP.CELSIUS = 10 TO 100
```

Now, press the End key (on the numeric keypad) and you'll see the cursor jump to one character past the end of the line. Type in the following addition to line 10:

```
10 FOR TEMP.CELSIUS = 10 TO 100 STEP 10
```

The STEP option of the FOR statement tells BASIC how we want it to change the value of *temp.celsius*. By adding STEP 10 to the FOR statement, we've told BASIC to add 10 (instead of 1) to *temp.celsius* (our loop counter) each time it bumps it at the end of the FOR loop.

List the program again if you want to be sure you made the change correctly. When you run the program this time, things will happen just as fast, but all of the output will fit on your screen rather than go zooming off the top of it!

There are few restrictions on the initial, target, or step values that can be used in FOR loops. You could, for example, use the following FOR statement to get the same output with a different twist:

```
10 FOR TEMP.CELSIUS = 100 TO 10 STEP -10
```

If you run the program with this FOR statement, *temp.celsius* will start off at 100 and decrease by 10 each time through the FOR loop. When BASIC sees a negative loop counter it reverses its behavior, and it stops executing the loop when *temp.celsius* reaches a value that is less than the target value of 10.

One restriction worth knowing about is that if the initial value of the loop counter exceeds the target value, the FOR loop won't be executed at all. For example, if you enter:

```
10 FOR TEMP.CELSIUS = 100 TO 10 STEP 20
```

and run the program, nothing will happen. The same thing is true if the initial value of the loop counter is less than the target when a negative STEP value is used. BASIC just doesn't bother doing things it doesn't have to do!

Experiment with different FOR loops and see where you get. For example, what happens when this FOR statement is used on line 10?

```
10 FOR TEMP.CELSIUS = 10 TO 10 STEP 10
```

or this one:

```
10 FOR TEMP.CELSIUS = 10 TO 100 STEP 1
```

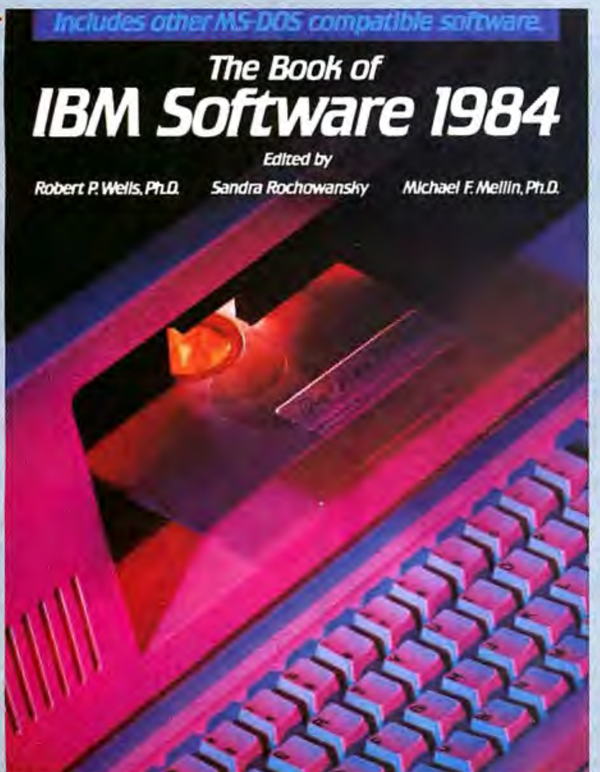
Can you see why BASIC executes these loops the way it does?

Once again, it would be a good idea for you to change the program to convert Fahrenheit temperatures to Celsius (save it as CONVERTF.BAS so you know which program is which). The new program should work in deferred mode and use a FOR loop. Experiment with the FOR loop to see what different initial, target, and step values for the loop counter do. Try also to create some BASIC programs of your own.

Next time we'll learn about the different kinds of numbers BASIC can work with and find out what those mysterious quotation marks are all about. See you then. ▲

DON'T BUY IBM SOFTWARE.

(UNTIL YOU READ THIS BOOK.)



THE BOOK OF IBM SOFTWARE 1984

Your IBM PC is only as good as the software that runs on it. So it makes sense to have the most detailed, up-to-date software review book there is. Especially when it can save you hundreds of dollars and hours of frustration.

The Book Company, the established authority in the field of software review guides, is famous for its consumer report styled evaluations.

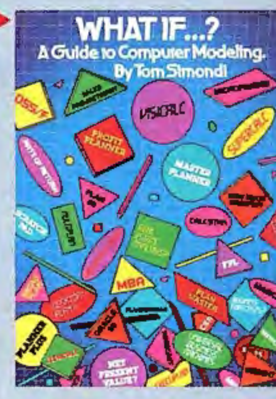
Our latest publication, **The Book of IBM Software 1984,** evaluates IBM PC/XT and IBM compatible software for such applications as Accounting, Database Management, Spreadsheet Modeling, Telecommunications, Word Processing, Education, Entertainment, Utilities and much more.

You get incisive, impartial evaluations written by experts. What's more, programs are reviewed and graded according to Ease of Use, Reliability and Value for Money. Plus you get all the hard facts such as price, system requirements, etc.

Don't settle for simplistic listings of available programs. Get ALL the information you need. Buy **The Book of IBM Software 1984** today.

THE BIG BONUS

When you send in the coupon and order **The Book of IBM Software 1984,** we'll send along, free of charge, our popular book on financial modeling, **WHAT IF...? A Guide to Computer Modeling** (retail value $19.95). If you buy **The Book of IBM Software 1984** from your dealer, send us proof of purchase and we'll send you the financial modeling book. Free.



TIPS ON BUYING SOFTWARE

Send for our free 64 page booklet, **"Tips on Buying Software".** You'll learn how to evaluate your software needs and find out what's available. Plus much more.

THE BOOK COMPANY
A Division of Arrays, Inc.

Available at computer stores everywhere or directly from The Book Company.

Title	Qty.	Price	Total
The Book of IBM Software 1984 (Includes **WHAT IF...? A Guide to Computer Modeling**)		$19.95	
Tips on Buying Software		N/C	

Mail to: THE BOOK COMPANY, Dept. STI, 11223 S. Hindry Avenue, Los Angeles, CA 90045. Or call toll free in California 800/441-2345, ext. 518. Outside California in the continental U.S. call 800/556-1234, ext. 518.

Name

Address

City State Zip

Visa, MasterCard, Check or Money Order accepted.

Card #

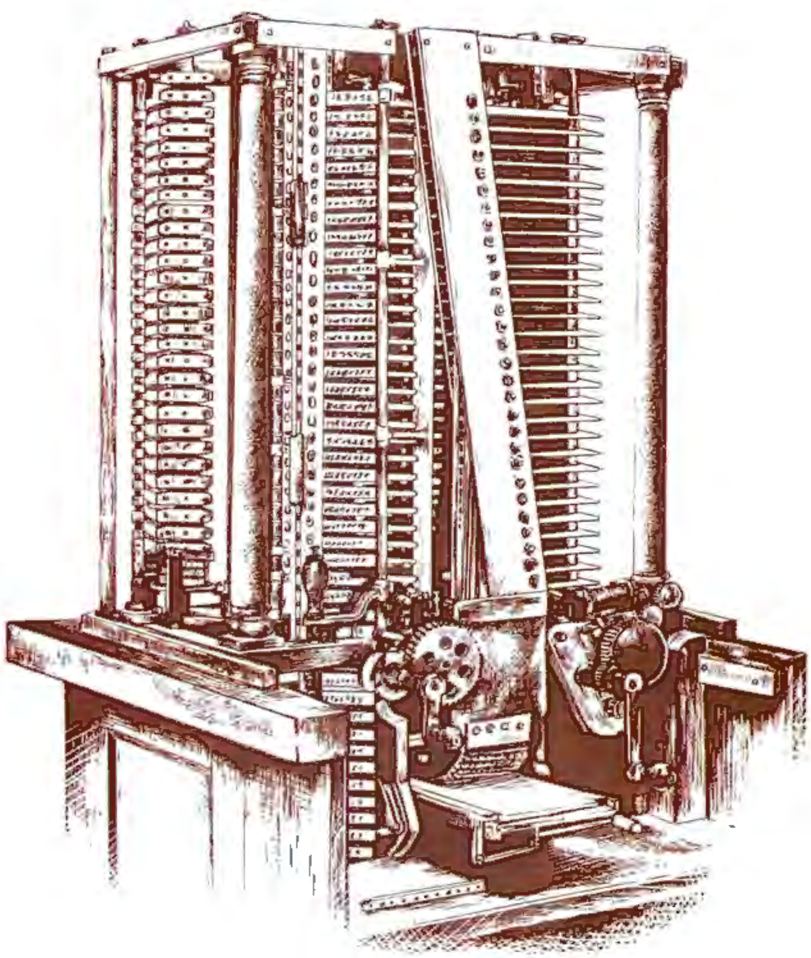
☐ Visa ☐ MasterCard Exp. Date Signature

Total Amount Enclosed $ California residents add 6% sales tax.

Postage will be paid by The Book Company.

IBM PC/PC XT are registered trademarks of International Business Machines Corp.

THE ANALYTICAL ENGINE



THE 8087 INSTRUCTION SET

BY ED BOGUCZ

A microcomputer and a number-crunching program usually make an odd couple in scientific and engineering computing. Until recently, only necessity or curiosity would send an analyst with a large-scale computing project to a micro. Because of memory limitations and lengthy execution times, microcomputer number crunching tested a programmer's skill, patience, and sanity.

New products, however, have helped make today's microcomputers better partners in number-crunching tasks. Floating-point processors now provide computing speeds comparable to those of many general-purpose minis and mainframes. And the old 64K memory barrier (a holdover from eight-bit machines) has been broken by several new compilers and interpreters (more on these in future installments of this series).

The chip that turns the PC into a real number cruncher is Intel's Numeric Data Processor (NDP), the 8087. Last month, we took a broad look at the 8087 and its significance. We discussed how the 8087 complements the PC's 8088 microprocessor by speedily handling all the time-consuming arithmetic operations. The 8087's powerful stack of eight eighty-bit registers was described, and the various data formats handled by the NDP were discussed.

This month, we'll conclude our introduction to the 8087 by

HEROISM IN THE MODERN AGE

THE ROLE-PLAYING GAME OF TODAY

- The first truly complete computerized role-playing system.
- A multi-character role-playing game which is vast in scope, allowing hundreds of incredibly detailed characters of your own creation.
- A sophisticated simulation of modern life and human psychology.
- A test of creative thinking and imagination.
- A game designed especially for the IBM PC*
- The only system realistic enough to be believable in *real-life* rather than a fantasy setting.

Your Starter Pack contains 2 full diskettes and detailed manuals. The game has 2 parts:

1. **BASIC MODULE** — which allows you to create characters and guide them through their lives. You can have the computer create characters — giving them names, a family background, and psychological and physical makeup — or you can custom-design characters, perhaps to represent people in real-life.

2. **PROJECT CONTACT Scenario #1** — which allows you to take characters from the Basic Module or pre-created characters on a secret mission to save the United States from a mad scientist and a terrorist plot. You must use all of your character's abilities and skills and technological tools to find clues, persuade allies, question suspects, fly helicopters and fight battles. You have only 60 hours to complete your mission...

HEROISM IN THE MODERN AGE is expandable, with other modules available.

MORE THAN JUST A GAME...
PROJECT CONTACT Scenario #1

at your computer store

SYSTEM REQUIREMENTS:
IBM PC with one double-sided or two single-sided disk drives, an 80-column monitor, and 64K RAM (96K for DOS 2.0).*
Color graphics/adaptor optional

IBM is a trademark of International Business Machines Corp.

Pacific Infotech Corp.

10850 Wilshire Blvd.,
Suite 800
Los Angeles, CA 90024
U.S.A.
(213) 470-7590

MUCH OF THE 8087'S SIGNIFICANCE LIES IN ITS METHOD OF OPERATION AND ITS POWERFUL INSTRUCTION SET.

looking at its instruction set. We'll present an assembly language program that will illustrate the simplicity of 8087 programming. To give you a feeling for the speed of the NDP, we've got some benchmark timings for this example program and equivalent versions in BASIC. We'll conclude this month's article with short reviews of two books about 8087 programming.

You don't need to be an assembly language whiz to follow this month's discussion. The primary intent is to get you better acquainted with the NDP's power. Much of the chip's significance lies in its method of operation and its powerful instruction set, and these topics are discussed best at the assembly language level. Detailed discussion of programming techniques, therefore, is left to the references.

INSTRUCTION SET

The 8087 instructions are listed in table 1 by functional group. The instruction mnemonics are Intel's. Note that Intel begins all its 8087 mnemonics with an *F* (for *floating-point*). This makes 8087 instructions easy to spot in 8088/8087 assembly language programs. Where more

Data Transfer Instructions:	
FLD/FILD/FBLD	Load (push) onto stack
FST/FIST	Store
FSTP/FISTP/FBSTP	Store and pop stack
FXCH	Exchange registers' contents
Arithmetic Instructions:	
Addition:	
FADD/FIADD	Add
FADDP	Add and pop stack
Subtraction:	
FSUB/FISUB	Subtract
FSUBP	Subtract and pop stack
FSUBR/FISUBR	Subtract reversed
FSUBRP	Subtract reversed and pop
Multiplication:	
FMUL/FIMUL	Multiply
FMULP	Multiply and pop stack
Division:	
FDIV/FIDIV	Divide
FDIVP	Divide and pop stack
FDIVR/FIDIVR	Divide reversed
FDIVRP	Divide reversed and pop
Other operations:	
FABS	Absolute value
FCHS	Change sign
FPREM	Partial remainder
FRNDINT	Round to integer
FSCALE	Scale
FSQRT	Square root
Transcendental functions:	
FPTAN	Partial tangent
FPATAN	Partial arctangent
F2XM1	2∧X−1
FYL2X	$Y*\log_2 X$
FYL2XP1	$Y*\log_2(X+1)$
Load constants:	
FLDZ	Load +0.0
FLD1	Load +1.0
FLDPI	Load pi
FLDL2T	Load $\log_2 10$
FLDL2E	Load $\log_2 e$
FLDLG2	Load $\log_{10} 2$
FLDLN2	Load $\log_e 2$
Comparison Instructions:	
FCOM/FICOM	Compare
FCOMP/FICOMP	Compare and pop stack
FCOMPP	Compare and pop twice
FTST	Test
FXAM	Examine
Processor Control Instructions:	
FINIT/FNINIT	Initialize processor
FDISI/FNDISI	Disable interrupts
FENI/FNENI	Enable interrupts
FLDCW	Load control word
FSTCW/FNSTCW	Store control word
FCLEX/FNCLEX	Clear exceptions
FSTENV/FNSTENV	Store environment
FLDENV	Load environment
FSAVE/FNSAVE	Save state
FRSTOR	Restore state
FINCSTP	Increment stack pointer
FDECSTP	Decrement stack pointer
FFREE	Free register
FNOP	No operation
FWAIT	CPU wait

Table 1. 8087 instruction set (Intel mnemonics).

than one mnemonic is listed for a given operation, the different mnemonics are used either to handle different data types (such as FLD/FILD/FBLD) or to achieve different processor operations (such as FINIT/FNINIT). For example, FLD is used to load real variables, FILD to load integers, and FBLD to load binary-coded decimals.

The instructions and their groups should be self-explanatory; the operations they represent are common to any scientific calculator. The only unusual aspects are the 8087's stack and some "reverse" operations. These topics are discussed briefly.

The principal programming advantage of the 8087 is that its registers can be used as a stack. Stack operations are familiar to any assembly language programmer and to anyone who has used a calculator with reverse Polish notation. On a scientific calculator, the Enter button causes the displayed value to be *pushed* onto the stack. Two-operand functions (such as addition and multiplication) typically involve the top two stack elements and result in the stack being *popped*, leaving the answer on top of the stack.

Values are pushed onto the the 8087 stack using various load instructions (such as FLD and FILD). The 8087 stack is popped by several different instructions (FSTP and FADDP, for example). Intel mnemonics for these instructions invariably end with a P.

Operands for NDP instructions typically are provided in the form (destination,source). For example, the instruction

```
FADD ST(2),ST(0)
```

means: Take the value in stack position 0 (top of stack), add it to stack position 2, and leave the result there (in stack position 2). Note that the Intel mnemonic for the stack position I is ST(I). In general, operands may be NDP registers or numbers stored in memory.

The reverse instructions perform operations that make for easier programming. For example, the division operation

```
FDIV ST(1),X
```

takes the value of X (from memory—in this example, the assembler knows the type of the variable), performs the division ST(1)/X, and leaves the result in ST(1). In contrast, the reverse operation

```
FDIVR ST(1),X
```

performs X/ST(1) before leaving the result in ST(1). Thus a programmer can conveniently compute either $A = A/B$ or $A = B/A$ (and $A = A-B$ or $A = B-A$).

BABBAGE LOG TABLE

An assembly language subroutine that illustrates the basics of 8087 programming is shown in listing 1. In honor of Charles Babbage, who created the original Analytical Engine, the subroutine computes a table of logarithms. (Babbage claimed that the process of manually calculating and checking log tables inspired him to design a computing machine.) Six lines of equivalent BASIC code are shown in listing 2.

The problem solved by the codes in listings 1 and 2 is straightforward: Calculate a table of log(X) (base e) for N values of X between X0 and X1. (The N values of X are to be equally spaced between X0 and X1 and to include the endpoints.) The codes must calculate the spacing

THE PURPOSE OF THE 8087 IS NOT EASE OF PROGRAMMING; IT'S PERFORMANCE.

between the X values (DX), as well as all the Xs. Note that the 8087 subroutine is called from a BASIC program. It finds the addresses of its arguments on the 8088 stack.

The heart of the 8087 assembly language program is the calculation loop Engine. Each pass through Engine computes the next log in the table. Note that at the start of every pass through Engine, the 8087 stack contains the same quantities: ST(0) = I, ST(1) = DX, and ST(2) = X0.

The loop proceeds much as one would continue with a hand-held calculator: First compute X(I), then log(X(I)); then increment I to get ready for the next pass. The simplicity of 8087 programming should be evident from the listing.

One minor point worth mentioning about the NDP code is the use of the 8087 instruction FYL2X, which calculates the quantity $Y * \log_2 X$. X is taken from the stack top and Y from ST(1). This operation facilitates the calculation of logs to bases other than two, since a multiplication is always required:

$$\log_n X = \log_n 2 * \log_2 X$$

The constants required for natural logs (base e) and common logs (base 10) are available in 8087 instructions (FLDLN2 and FLDLG2 respectively).

BENCHMARK TIMES

The purpose of the 8087 is not ease of programming; it's performance. Any extra programming effort (or 8087 software package) must be justified by benefits of faster execution, increased accuracy, or both.

For purposes of comparison, our 8087 example subroutine was run and timed along with interpreted and compiled versions of the BASIC code. To give some semblance of realism (and also to check accuracy), the subroutines duplicated the log tables that appear in the analyst's bible, *Handbook of Mathematical Functions* (edited by Milton Abramowitz and Irene A. Stegun, Dover Publications, 1965). The log tables

```
;            SUBROUTINE BABBAGE(X(Ø),F(Ø),N,XØ,X1)
;            Subroutine for computing table of (natural) logs.
;            Calculates log(X) for N values of X between XØ and X1.
;
;            Subroutine calculates X(I) values, I=Ø,N-1, and stores them
;            in array X; LOG(X(I)) values stored in array F. Arrays X and F
;            must have dimension of at least N. Listing shows source for
;            single-precision X and F; changes for double precision are noted
;            in comments.
;
;Begin code:
             PUBLIC    BABBAGE
CSEG         SEGMENT   'CODE'
             ASSUME    CS:CSEG
BABBAGE      PROC      FAR
             PUSH      BP                    ;Save BP
             MOV       BP,SP                 ;Get 8088 stack pointer
;
;Initialize array indexes:
;            During calculations, register BX will be used as an index for
;            array X; DI will be used for F. Here, load array starting addresses.
             MOV       BX,[BP]+14            ;Address X(Ø)
             MOV       DI,[BP]+12            ;Address F(Ø)
;
;Initialize NDP and move XØ and X1 onto stack:
             FINIT                           ;8087 initialization
             MOV       SI,[BP]+8             ;Address XØ
             FLD       DWORD PTR [SI]        ;Push XØ onto 8087 stack
                                             ; 'DWORD PTR' for short real here
                                             ; 'QWORD PTR' for double precision
             MOV       SI,[BP]+6             ;Address X1
             FLD       DWORD PTR [SI]        ;Push X1 onto 8087 stack
;
;Calculate DX=(X1-XØ)/(N-1):
             FSUB      ST(Ø),ST(1)           ;Perform X1-XØ, leave on top of stack
             MOV       SI,[BP]+1Ø            ;Address N
             FILD      WORD PTR [SI]         ;Push N onto 8087 stack
                                             ; ('WORD PTR' indicates word integer)
             FLD1                            ;Load 1
             FSUBP     ST(1),ST(Ø)           ;Subtract and pop, leaving N-1 on top
             FDIVP     ST(1),ST(Ø)           ;Divide and pop, leaving DX on stack top
;
;Initialize counters CX (in 8088) and I (in 8087):
             MOV       CX,[SI]               ;Move N into CX for loop counting
             FLDZ                            ;Load zero
;
;Calculation loop:
ENGINE:
;            At start of loop, 8087 stack contains:
;                      ST(Ø) = I,    ST(1) = DX,    ST(2) = XØ.
;
;Calculate X(I):
             FLD       ST(Ø)                 ;Bump I into stack again
             FMUL      ST(Ø),ST(2)           ;Multiply by DX
             FADD      ST(Ø),ST(3)           ;Add XØ
             FST       DWORD PTR [BX]        ;Store X(I)
;
;Calculate LOG(X(I)):
             FLDLN2                          ;Push LOGe(2) onto stack
             FXCH                            ;Exchange ST(Ø),ST(1)
             FYL2X                           ;Calculate ST(1)*LOG2(ST(Ø))
                                             ; = LOGe(X(I)) and pop stack
             FSTP      DWORD PTR [DI]        ;Store log and pop stack, I now on top
;
;Increment counter and indexes:
             FLD1                            ;Load 1
             FADDP     ST(1),ST(Ø)           ;Add 1 to I and pop stack so I on top
             ADD       BX,4                  ;Add to array indexes:
             ADD       DI,4                  ;4 for single precision, 8 for double
;
;Repeat 'til finished:
             LOOP      ENGINE
;
;Return:
             FWAIT                           ;8088 wait for 8087 to finish last store
             POP       BP                    ;Restore BP
             RET       1Ø
BABBAGE      ENDP
CSEG         ENDS
             END
```

Listing 1. Example 8087 subroutine.

```
900 ' BABBAGE LOG TABLE:
910 '     Compute table of log(X) for N values of X between XØ and X1
920 '     Store X values in array X(I) and log(X) values in F(I).
930 '
940 DX = (X1 - XØ) / (N-1)        ' X spacing
950 FOR I = Ø TO N - 1            ' Corresponds to loop ENGINE in 8087 code
960      X(I) = X(Ø) + I * DX     ' Calculate X(I)
970      F(I) = LOG(X(I))         ' Calculate log(X(I))
980 NEXT I
990 RETURN
```

Listing 2. Equivalent BASIC subroutine for example problem.

IN MATRIX MULTIPLICATION THE 8087 BEATS THE BASIC INTERPRETER BY A FACTOR OF ABOUT 160...

in the handbook list results (to sixteen figures) for 2,100 values of X between 0.001 and 2.100.

Table 2(a) lists execution times of the calculations for both single- and double-precision results. Also listed are the ratios of execution time for the Basic codes to the 8087 subroutine. (The times listed in table 2 are accurate to the number of figures quoted. To get accurate timings, each subroutine was executed several times so that any inaccuracy in the total time measured would not affect the single-execution time, at least to the number of digits reported.)

Any number-cruncher will agree that the results in table 2(a) are impressive: A simple 8087 code cuts execution time by a factor of fifteen or more compared with compiled Basic code, and by a factor of fifty or more for interpreted Basic. In this example, the speed of the 8087's internal log calculation gives it a special edge over non-8087 code, particularly for double-precision results. In general, applications that involve many transcendental function evaluations can expect similar results.

The results in table 2(a) illustrate a point made last month: The 8087 is especially significant in applications that require double-precision (sixty-four-bit) results. Indeed, programs that use only single-precision (thirty-two-bit) numbers really don't exploit the power of the NDP; in a sense, these programs throw away computed results.

Note that the time required by the 8087 Babbage code is essentially equal for single- and double-precision results. This is because the 8087 performs all calculations in its internal eighty-bit temporary real format. The external format of a variable is usually of no significance to the NDP once the variable has been loaded. Hence execution times for the fundamental operations generally aren't affected by the type of the variables in memory (multiplication is the only exception). The only difference in execution time between single- and double-precision variables (usually) is the load and store times. Essentially, the difference is between the time required to read/write four bytes or eight bytes. For most applications this isn't significant.

In contrast, interpreters and compilers that use general-purpose processors like the 8088 generally take much longer to calculate double-precision than single-precision results. Note that the interpreter's time for double-precision logs is more than four times greater than single-precision results, and the compiler's time more than doubles.

(It should be noted here that only the Basic 2.0 interpreter can compute double-precision transcendental functions accurately. Although Basic 1.1 might seem to return a double-precision result in a log evaluation, only the first six digits are accurate. It uses a single-precision log routine and then converts the result to a double-precision variable by putting junk in the remaining four bytes.)

	Basic Interpreter (T1)	Basic Compiler (T2)	8087 Assembly (T3)	T1/T3	T2/T3
Babbage log table (2,100 logs)					
Single precision	35.6	11.0	0.681	52	16
Double precision	155.9	28.6	0.695	224	41
20 x 20 matrix multiplication					
Single precision	83.6	7.6	0.497	168	15
Double precision	96.3	11.7	0.598	161	20

Table 2. Benchmark time results (in seconds).

MATRIX MULTIPLICATION BENCHMARK

Calculating a log table is an admittedly esoteric exercise. Benchmark results for this problem might seem to have little bearing on everyday 8087 applications. To get a better feeling for 8087 performance, look at the execution times for typical matrix multiplication routines in table 2(b). The text exercise is the multiplication of two twenty-by-twenty matrixes. The problem requires eight thousand multiplications and eight thousand additions.

The results of the matrix multiplication test show that the 8087's performance on the log table problem was fairly representative of its capabilities. In matrix multiplication, the NDP beats the Basic interpreter by a factor of about 160, and the compiler by at least fifteen.

Notice that the difference between the 8087's single-precision and double-precision times is much greater on the matrix multiplication test than it is on the log table. This is because the NDP executes a multiplication much faster when one operand is a single-precision real.

REFERENCES

Intel provides some good references on the 8087. The *iAPX86/88, 186/188 User's Guide* has an excellent chapter on the 8087. This chapter contains good discussions about the general concepts of floating-point processing in addition to specific details about the 8087, and it's a valuable reference for anyone interested in computer operations. In addition, a good application note, "Getting Started with the Numeric Data Processor," is available free from Intel. The note contains comments on and examples of 8087 use and programming that will be of interest to serious assembly language programmers.

In addition to publications by Intel, two books about 8087 programming are currently available. *8087 Applications and Programming for the IBM PC and Other PCs*, by Richard Startz, is a good book for beginning 8087 users and programmers. The author provides general background about the chip, a description of each 8087 instruction, an introduction to assembly language programming, and a host of programming examples that address some fundamental numeric techniques of scientific and engineering analysis. The emphasis is on developing useful subroutines that can be called from Basic programs. Standalone assembly language programs are beyond the book's scope.

A desire to master the details of assembly language programming is not required for profitable use of *8087 Applications and Programming*. The programs presented in the book are available on an optional disk. The disk provides a storehouse of fundamental routines in ready-to-use format and, priced at $30, is a bargain for beginning programmers.

Startz's title is somewhat misleading. The key word is *applications*. The book is packed with the theory and practice of some fundamental

25 Minutes' Worth of Reading.

Cut your reading time in half or better — and improve your comprehension with SPEED READER II.

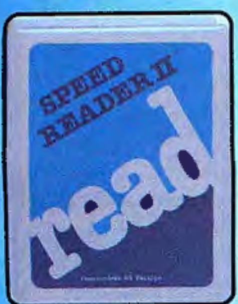
This Award-winning software was designed by Reading Specialists and has proven effective for thousands.

SPEED READER II is fun, stimulating ... and best of all, it gets results!

teacher certified NEA

Compatible with Apple, IBM, PC Jr. and Commodore 64.

Price only $69.95



Davidson.

Educational Software that Works.

6069 Groveoak Place, #12
Rancho Palos Verdes,
CA 90274 (213) 373-9473

MACRO PROGRAMMING IS AN EXTREMELY USEFUL TECHNIQUE OF THE ADVANCED PROGRAMMER...

matrix operations. It might well be called *Introductory Numerical Methods for Scientists and Engineers, with Programs for the 8087.* Startz introduces each method with background information about its significance. Some readers may even feel that there is too much introductory material—the sections entitled "What is a matrix?" and "Why are matrices interesting?" for example—and not enough comment on advanced methods, programming techniques, or implementation problems.

Even so, Startz's book provides a clear and basic introduction to 8087 programming. Beginning assembly language programmers will find the text straightforward and useful. 8087 programs are always introduced with equivalent Basic codes so that readers can compare the two.

The programs in Startz's book are significant in their own right. They range from simple applications, such as summing a list of numbers, to copying arrays or evaluating transcendental functions. Also included are routines for vector inner product, matrix multiplication, matrix inversion, and Crout decomposition (a method for solving linear equations). Routines for converting between Microsoft and IEEE floating-point formats are also provided. (Floating-point format conversion is a nuisance in practice; more on this problem in our next article.)

For the user's convenience, the routines are provided on the disk in three forms: as an assembly language source program, as an assembled object module, and as a file ready to be bloaded into an interpreted Basic program. The assembly language source code is in a form that can be processed directly by the IBM *Macro Assembler;* 8087 mnemonics (which the IBM assembler doesn't recognize) have been recoded in the source into instructions that the IBM assembler can process. The object-module versions of the routines are ready for linking with user-supplied code. The *bsave* versions make for quick and easy use of the routines with interpreted Basic programs, and the provision of the codes in this form is convenient for the beginner.

As an application of the 8087 routines developed in his book, Startz presents a simple, general-purpose statistical analysis Basic program. The program performs multiple regression and other statistical calculations. It runs interactively and includes modules for data entry, editing, and analysis. Other Basic programs on the disk perform Crout decomposition, numeric integration and differentiation, and the solution for nonlinear equations.

8087 Applications and Programming is valuable for two different audiences. First, for Basic programmers who have always been tempted to learn assembly language, the book makes the plunge painless. Second, for people who don't intend to become proficient 8087 assembly language programmers, Startz provides some general routines that can be immediately applied in user application programs.

87/88 Guide, by Stephen Fried, really isn't a book at the moment. It's a book in the making—and a good one, too. Fried is the primary author of much of the software developed by MicroWare, a firm specializing in 8087 support. He is finishing a book on 8087/8088 assembly language programming for publication sometime later this year. MicroWare bundles copies of Fried's latest draft (in loose-leaf form) with some of its products. It also is available separately. Naturally, much of the book discusses MicroWare products. This review will consider only the general aspects of the book. MicroWare's products will be reviewed next time.

The *87/88 Guide* is a comprehensive look at assembly language programming for the 8088 and 8087. Unlike Startz, Fried emphasizes programming techniques and standalone assembly language programs. In a sense, he picks up where Startz leaves off. The book provides a wealth of detailed information and programming tips suitable for the serious assembly language student or the experienced programmer.

Included are sections on the basics of assembly language programming, 8088 and 8087 instructions, and interfacing assembly language programs with IBM DOS and BIOS. The writing of assembly language macros is also covered in considerable depth. Macro programming is an extremely useful technique of the advanced programmer, and many examples are given to show its power.

The book also helps the reader develop standalone assembly language programs, and several examples are presented. One utility loads any file from disk to a designated memory location.

Fried's book has two special attractions. First, example programs are well commented. The comments are so frequent and detailed that they give the reader the feeling of a personal guided tour through the code that's being presented. (As a bonus, the comments frequently are entertaining as well, especially when Fried exposes the "dirty tricks" of experienced programmers.) Second, Fried presents some powerful macros that are useful in 8087 programming. These include routines that provide higher-level functions for the 8087.

COMING SOON

Readers who have made it this far will probably be looking for information about assembling 8088/8087 code. Sorry, you'll have to wait till next month.

As noted previously, the IBM *Macro Assembler* won't process 8087 mnemonics directly. Next month, we'll review the various products available for assembly of 8088/8087 code. In addition, we'll look at some packages that allow IBM compilers to drive the 8087 directly. ▲

8087 Applications and Programming for the IBM PC and Other PCs, by Richard Startz. Robert J. Brady Company (Bowie, MD 20715; 800-638-0220). $19.95 (Optional disk, $30).
87/88 Guide, by Stephen S. Fried. MicroWare (Box 79, Kingston, MD 02364; 617-746-7341). $30.
Getting Started with the Numeric Data Processor, Application Note AP-113. Intel Corporation (Literature Department SV3-3, 3065 Bowers Avenue, Santa Clara, CA 95051; 800-538-1876). No charge.
iAPX 86/88, 186/188 Programming User's Guide. Intel Corporation (Literature Department SV3-3, 3065 Bowers Avenue, Santa Clara, CA 95051; 800-538-1876). $16.95.

Compute and print client tax returns in minutes

on your microcomputer with
*MICRO-TAX**

That's right. In just minutes you can have a client's completed tax return in your hand. Think about it — you increase client volume, you increase your profits. Plus, you save the cost of your computer service bureau—and you have complete client security.

FEDERAL AND STATE PACKAGES TO MEET EVERY NEED. MICRO-TAX* offers four Federal tax packages and 25 state packages (fully integrated with the Level II Program), so you can select the programs that best meet your needs:

Level I—Federal Individual Package: for individuals preparing their own taxes.
Level II—Federal Professional Individual Package: for accountants, registered agents, tax attorneys, and other tax professionals.
Level III—Federal Partnership/Corporate Package: for those who prepare Federal Partnership, Corporate, and Subchapter S returns.
Level IV—Overseas Tax Package: addresses the unique tax situations of United States Expatriates.
Levels II, III, and IV have a depreciation module and automatically compute underpayment penalties and minimum tax. In addition, Levels II and III automatically compute self-employment taxes, and Level II computes income averaging.

FLEXIBLE DATA ENTRY. With MICRO-TAX* you can organize data entry in a sequence similar to that of manual tax preparation, or you can choose another sequence. The menu driven system makes data entry simple.

MULTIPLE PRINTING OPTIONS. You can input client tax information at the time of interview and produce forms immediately, or enter data during the day and batch print returns at night. MICRO-TAX* prints your returns on IRS forms, IRS approved substitute forms, or with transparent overlays.

TAXNET*—TELETEXT SUPPORT NETWORK. MICRO-TAX* customers can now have access to an electronic mailbox and instantaneous memoboard through the TAXNET* teletext support network. With TAXNET,* you can send information, ask questions, get answers and updates—directly through your computer and a modem.

TAX ORGANIZER. Now MICRO-TAX* offers a Tax Organizer. You get both the software and the forms, so each year, you can send your clients an organizer with the prior year's client data printed on it.

HARDWARE COMPATIBILITY. MICRO-TAX* is compatible with your IBM PC/XT,* DEC Rainbow,* Radio Shack,* or any other personal computer with CP/M-80,* PC DOS,* or MS DOS*—from Apple* to Zenith.*

So, take the tedium out of tax preparation—save time and money—
Call Micro-Tax for complete details, or call your local dealer.*

1983 FEDERAL FORMS AND SCHEDULES INCLUDED

FULL FEDERAL MICRO-TAX* PERSONAL COMPUTER SYSTEMS	1040, 1040A, 1040ES	1065, 1120, 1120S	A, B, C, E, G, SE, W	D, F	K, K-1	R, RP	1116	2106	2119	2210	2220	2440	2441	2555	3468	3903	3903F	4137
Level I, Individual $195.	•		•	•				•					•					
Level II, Professional Individual $1000.	•		•	•		•	•	•	•	•		•	•		•	•		•
Level III, Partnership/Corporate $1000.		•		•	•						•				•			
Level IV, Overseas $2000.	•		•	•			•	•		•				•		•	•	

FULL FEDERAL MICRO-TAX* PERSONAL COMPUTER SYSTEMS	4255	4562	4626	4684	4797	4835	4952	4972	5695	6251	6252	90-22.1	ACRS	BATCH	PRINTS: Substitute Forms	PRINTS: IRS Forms	PRINTS: Blank Paper	PRINTS: Continuous Preprinted Forms
Level I, Individual $195.									•						•	•	•	
Level II, Professional Individual $1000.	•	•		•	•	•	•	•	•	•	•		•	•	•	•	•	•
Level III, Partnership/Corporate $1000.	•	•	•	•	•	•					•		•	•	•	•	•	•
Level IV, Overseas $2000.					•	•				•		•	•	•	•	•	•	•

• All forms and schedules subject to final IRS changes for 1983 tax year.
• 25 integrated state returns available • Orders taken for yearly update packages
• Transparencies: Level I, $150; Levels II, III, IV, $250 each.

*C/PM—trademark of Digital Research, Inc.; DEC Rainbow—trademark of Digital Equipment Corp.; MICRO-TAX and TAXNET—trademarks of Microcomputer Taxsystems, Inc.; MS DOS—trademark of Microsoft Corp.; PC DOS, IBM PC, and IBM XT—trademarks of IBM; Apple—trademark of Apple Computers; Zenith—trademark of Heath Company and Zenith Radio Corp.; Radio Shack—trademark of Tandy Corp.

MICRO-TAX* MICROCOMPUTER TAXSYSTEMS, INC.
6203 Variel Avenue, Suite A Woodland Hills, CA 91367 Dept. 1I Phone (213) 704-7800
(Area code changes to (818), effective January 1984)

THE C SPOT

by Rex Jaeschke

Defining and Calling Functions

This month we will learn how to define and call functions, and we'll see the various mechanisms used in passing arguments between them.

Functions. Some languages allow data to be passed back from subroutines via arguments, while others allow the subroutine name itself to contain a returned value. FORTRAN and BASIC allow both methods. C, however, allows a subroutine to do either, both, or neither. In C, all subroutines are called functions, and a function does not need to return any value or have any arguments.

```
/* funct.c Demonstrate passing data between functions */

main ()
{
        printf ("Length of string is %d.\n",strlen ("My name is John"));
}
#define NULL '\0'

strlen (string)
char string[];
{
        int i;

        i = 0;
        while (string[i] != NULL)
                ++i;
        return (i);
}
```

produces

```
Length of string is 15.
```

main invokes two functions, *printf* and *strlen*. *printf* is contained in the compiler vendor library, and *strlen* is contained in the same source code file as *main*. (*strlen* generally is supplied too.) C allows functions to be compiled together or separately. Functions may not be nested.

strlen has only one argument, the string whose length is to be determined. *strlen* returns an integer length that is used in place of the function call as the second argument to the *printf* function. In this example, *strlen* is like the *len(X$)* function in BASIC. For array arguments, the argument address is passed to the function rather than the argument value. This is termed *call by reference*. We have seen that literal text strings are treated by the compiler as character arrays; therefore the address of the text string (and not its value) is passed to *strlen*.

It is the programmer's responsibility to make sure that the number and type of arguments passed to a function match those expected by that function. Unfortunately, the number of arguments passed is not stored or made available to the called function. *printf* seems to be able to handle a variable number of arguments. This is because the first argument contains a number of display edit masks, one for each subsequent argument. Users of *printf* must ensure that the number and type of edit masks match the remaining arguments. Experiment with *printf*, specifying too many or too few arguments for the edit mask. Results will be unpredictable.

In *strlen*, the arguments must be declared appropriately, so their type will be known. No storage is allocated for the dummy arguments within *strlen*. The argument list is specified between the parentheses after the function name declaration. The argument types must be declared immediately after the argument list and before the function's opening brace. For character arrays such as *string[]*, the dimension isn't needed, as the string's address was passed and the string array ends when a null is found. C performs no array bounds checking at run time.

The variable names *string* and *i* used by *strlen* are local to *strlen* and are inaccessible to all other functions, including *main*. The length that *strlen* computes is returned to *main* by the *return* statement. The value

Come in late on "The C Spot"? All back issues of the column—from January 1984—are still available; for further information, see page 4.

in the parentheses can be an expression. A function need not return a value; if it doesn't, the form *return;* is used instead of *return (rval);*. If a function doesn't contain a *return* statement, the final closing brace acts as a *return;*. If no expression or value is specified with the *return* statement or there is no *return* statement, an undefined value is returned. Therefore, don't try to make sense out of the value returned by a called function if it doesn't return one explicitly.

Note that the *while* construct properly handles strings of all lengths, including one containing only a null. Consider the case where the string passed to *strlen* is missing its terminating null. *strlen* will continue incrementing *i* until it finds a null. In doing so, the value of *i* may overflow; that is, *i* may be incremented until it exceeds its maximum possible value. On sixteen-bit twos' complement machines, such as the PC, the maximum possible size of *i* is 32,767. Numeric overflow might cause the program to abort; or, worse, *i* might roll over to a negative value. If the latter were to happen, incrementing 32,767 by 1 would give *i* a value of −32,768—a very strange length indeed.

Variable *i* in *strlen* could be declared as an unsigned integer allowing for a maximum length of 65,535, assuming that the unsigned field is stored in sixteen bits. Try writing a version of *strlen* called *lstrlen* that returns the string length as a long integer.

Run the following program several times. It should return different lengths for *string*, depending on what garbage is present in memory when *string* is declared. Any element of *string* (other than the last) containing a binary zero (\0) will cause the *strlen* function to terminate prematurely, thus giving a length other than 1,000. *array2.c* uses the *strlen* function defined above.

```
/* array2.c      Demonstrate the need for array initialization.
                 Failure to initialize the array string may
                 cause strlen to return an unexpected result */

main ()
{
        char string[1001];
        int i;

        string[1000] = '\0'; /* add end-of-string */
        printf ("Length of string is %d\n", strlen (string));
}
```

Bugs like this are difficult to locate, so it's crucial that all character arrays be properly initialized and terminated.

Calling by Value. In C, all simple arguments (nonstructures) are passed by value; that is, their value (rather than their address) is passed to the called function. Called functions treat arguments passed by value as automatic variables.

```
/* funct1.c      Demonstrate calling by value
                 THIS PROGRAM WILL NOT PRODUCE THE DESIRED RESULT */

main ()
{
        int i;

        for (i = 1; i <= 5; i++) {
                sub (i);
                printf ("%d\t",i);
        }
}

sub (n)
int n;
{
        n *= 5;                 /* same as n = n * 5  */
}
```

produces

```
1       2       3       4       5
```

rather than the expected

```
5
```

When *i* is used as an argument to the function *sub*, its value is passed, *not* its address. As *sub* does not know the address of *i*, it cannot alter its value. It can modify only its own private, temporary copy of that variable.

You can alter the value of an argument, provided it was passed by address rather than by value. This involves the use of pointers and will be discussed in later installments.

Calling by Reference. Arrays and other structure-type arguments are always passed by reference (address), and their elements can be modified by the called function. The function *strlen* used above has one argument that is a character array. This array address is passed to *strlen*.

The following function, *compress*, has one argument that is a character array. As the address of the string is passed, *compress* can legitimately modify the elements of the string. It does so by squeezing out spaces and tabs and adding a new end-of-string terminator.

```
#define EOS '\0'
#define SPACE ' '
#define TAB '\t'
```

COULD YOU ANSWER THESE QUESTIONS

1. *What is the latest R&D activity in Japan in the field of industrial robots?*

2. *Is the name "datascan" trademarked?*

3. *Which drugs have been successfully used to treat osteoporosis?*

4. *Where has Bank of America been advertising lately?*

5. *What is the current and projected market for frozen orange juice?*

6. *Are there any personal computer user groups for attorneys in the Boston area?*

Introducing In-Search, a simple software package that gives you almost immediate answers to literally millions of questions.

Complete, up-to-date information is a powerful advantage in any profession—information about research, technology, and competitors; information about trends, markets, products, and people.

If information was more accessible, you could have all the facts you need before you make important decisions. And the more facts you have, the better your decisions will be.

Answers.

How do you find the answers to your questions? In-Search.

In-Search is an extremely easy-to-use personal computer software package that lets you retrieve, instantly, answers to your questions from the wealth of knowledge stored in over 200 computer databases.

These databases are huge collections of printed material that has been computerized. With In-Search, your IBM (or compatible) personal computer and modem, you can reach this information over your own phone lines.

All The Answers.

Imagine scanning thousands of documents in seconds. In-Search can do just that.

In-Search allows you to read, study and use over 80 million articles and references from thousands of sources: Newspapers, magazines, technical journals, wire services, periodicals, annual reports and Yellow Pages.

You will be able to access information on all aspects of hundreds of subjects, including:

- Engineering and Science (complete coverage of literature since 1970)
- Medicine (complete coverage of literature since 1966)
- Business
- Government
- Current Affairs
- Arts and Social Sciences

All 200 databases are centralized at DIALOG® Information Services, Inc., the world's largest collection of online database information.

Easy Answers.

Before In-Search, finding information even via computer was tedious and time-consuming.

Not anymore. Just type into the computer the words or subject you'd like to research and In-Search takes it from there, bringing you articles or references containing the information you need. Simple, on-screen graphics guide you through every step.

```
compress (string)
char string[];
{
        int i,j;

        for (i = 0, j = 0; string[i] != EOS; i++)
                if (string[i] != SPACE && string[i] != TAB)
                        string[j++] = string[i];

        string[j] = EOS;
}
```

Functions Returning a Noninteger Value. By default, functions using the *return (rval);* statement return an integer value. To return other than an integer, both the calling and the called function must be notified. The *strlen* function discussed earlier returned an integer value by default. Strictly speaking, our example *funct.c* should be written as

```
/* funct2.c        funct.c - revised.
                   Use explicit type declaration for function strlen. */
main ()
{
        int strlen ();                  /* indicate that strlen returns an int */

        printf ("Length of string is %d.\n",strlen ("My name is John"));
}
#define NULL '\0'

int strlen (string)                     /* direct strlen to return an int value */
char string[];
{
        int i;

        i = 0;
        while (string[i] != NULL)
                ++i;
        return (i);
}
```

In *main*, the function *strlen* is declared explicitly to return an integer value. When function *strlen* is defined, it has a prefix of *int*, which tells *strlen* to return an integer value. For integer function values, these type declarations can be and usually are omitted. *funct3.* shows a function returning a double-precision floating point value.

```
/* --- funct3.c Function returning a double value --- */

main ()
{
        double i,sub ();                /* indicate that sub returns a double value */

        for (i = 1.1; i < 2.0; i += 0.2)
                printf ("i = %f\tsub (i) = %f\n",i,sub (i));
}
double sub (j)                          /* direct sub to return a double value */
double j;
{
        return (j * 3.1415926);
}
```

Output produced is

```
i = 1.100000    sub (i) = 3.455752
i = 1.300000    sub (i) = 4.084070
i = 1.500000    sub (i) = 4.712389
i = 1.700000    sub (i) = 5.340707
i = 1.900000    sub (i) = 5.969026
```

Any data type prefix can be used with a function definition to dictate the type of the value returned. If *sub* is an *int* function, then *return (1.234);* causes *sub* to return a value of 1. The result of the return expression is converted to the type of the function before the function terminates. The programmer must ensure that the value returned and the value expected are of the same type.

In expressions and function calls, *char* values are converted to, and hence are treated as, *int* values. For example, if *abc* is a *char* variable,

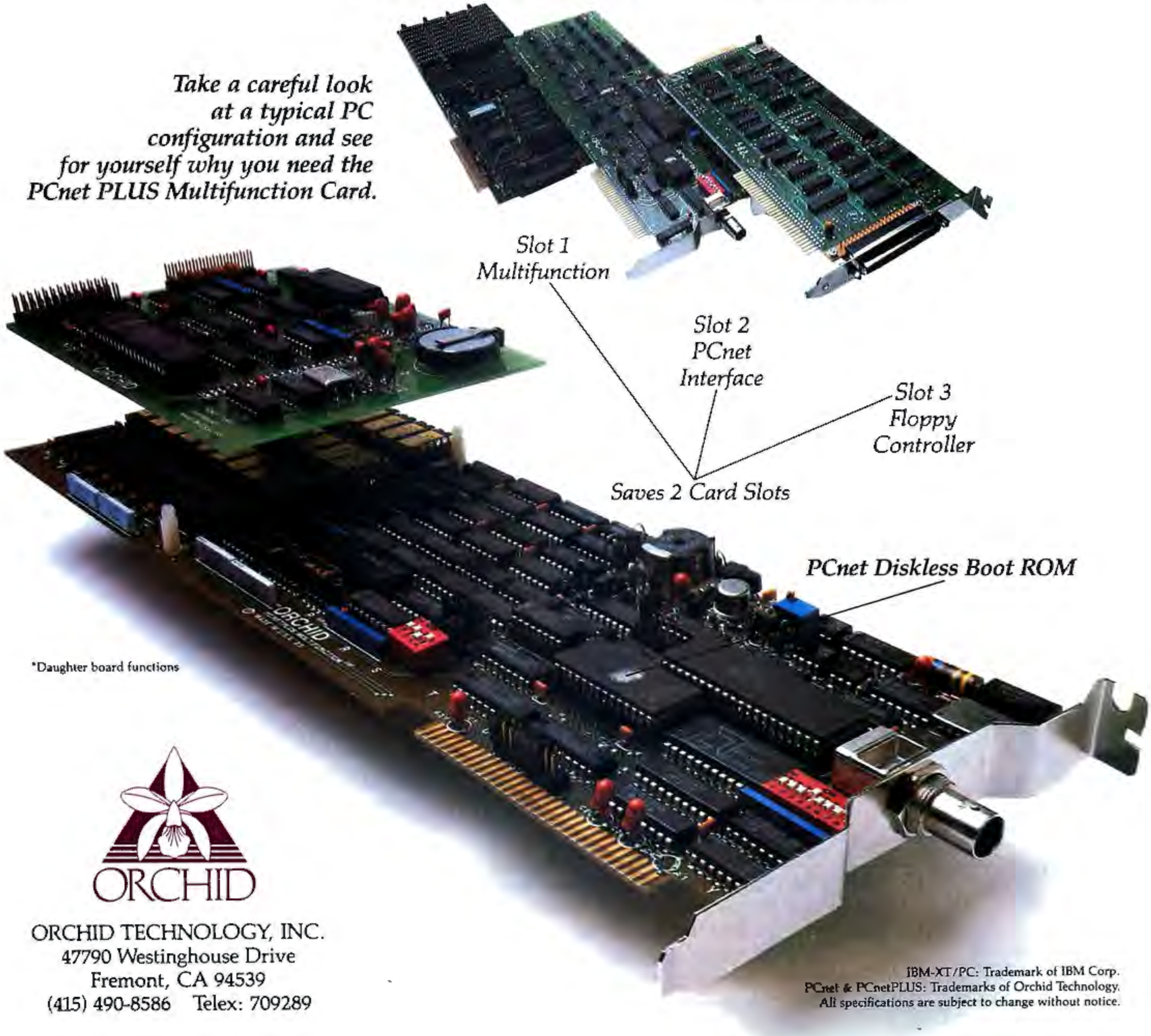
PCnetPLUS™ Multifunction Board from ORCHID

Go with Expandability

Integrating the leading low cost network, PCnet™ with the most popular add-on board

- ▲ **PCnet Interface**—The leading Local Area Network for the IBM-XT/PC™ and compatibles.
- ▲ **PCnet Software**—Allows PCs to share data, disks, printers, etc.
- ▲ **RAM Expansion Sockets**—Up to **256K** of memory.
- ▲ **PCnet RAM Drive**—High speed disk emulator software.
- ▲ **PCnet SPOOL**—Print Spooler software.
- ▲ **EPROM Socket**—For ORCHID supplied **floppyless PCnet** capability.
- ▲ **Serial Port***—COM1: or COM2:
- ▲ **Parallel Port***—LPT1: or LPT2:
- ▲ **Clock/Calendar***—With battery back-up and software.

Take a careful look at a typical PC configuration and see for yourself why you need the PCnet PLUS Multifunction Card.



*Daughter board functions

ORCHID

ORCHID TECHNOLOGY, INC.
47790 Westinghouse Drive
Fremont, CA 94539
(415) 490-8586 Telex: 709289

then *(abc — '0' + 6)* is an expression involving three integers. *sub (abc);* invokes function *sub* with a *char* argument. The value of *abc* is passed to *sub* as an *int* value. Likewise, functions returning *char* values do so in the form of integers. Hence, there is no need to declare explicitly the type of *char* functions, although you might do so for documentation.

The *%f* edit mask in *printf* above is used to print float and double values in the form *[—]mmm.nnn.* The default form, as used, provides for six fractional digits.

When a prefix is used with the function definition, that definition will be harder to see in a source code file. If you must have more than one function in a source file, you should insert a blank line or some obvious function delimiter such as a comment line filled with asterisks between successive functions.

Recursion. A function that calls itself directly or indirectly is said to be *recursive*. Like Pascal, C allows recursive calls. Recursive code is usually easier to read and write than nonrecursive code, although it probably doesn't increase efficiency or decrease storage requirements. Most programmers will never need to write recursive code. However, it is commonly used for implementing B-tree structures in sorting and searching algorithms.

Many programmers get confused when introduced to recursion. It really doesn't involve any magic, though. If you have trouble understanding the *outdec* function below, work through an example with pencil and paper until you are convinced that you understand it.

```
/* — recurse.c recursion example
        print integers using both printf and the recursive
        function outdec */
main ()
{
        int i;

        printf ("outdec\tprintf\n\n");
        for (i = -150; i <= 150; i += 50) {
                outdec (i);
                printf ("\t%d\n",i);
        }
}
outdec (number)
int number;
{
        int val;

        if (number < 0) {
                putchar ('-');
                number = -number;
        }
        if ((val = number/10) != 0)
                outdec (val);
        putchar ((number % 10) + '0');
}
```

produces

```
outdec  printf

-150    -150
-100    -100
-50     -50
0       0
50      50
100     100
150     150
```

Once *outdec* knows that it has a negative integer to print, it displays a minus sign and thereafter treats the number as positive. This will cause a problem on twos' complement machines when the largest negative integer (—32,768 on the PC) is to be displayed. Because that number has no positive equivalent, the statement *number = —number;* would produce an unpredictable result. Although *outdec* has a limitation, it ably demonstrates recursion.

Unfortunately, binary-to-character display conversion routines such as *outdec* make the low-order digits available before the high-order ones, although they must be printed in the opposite order. Therefore, the low-order digits must be stored for later use. Each time *outdec* is invoked, a different automatic variable *val* is created on the stack. % is the modulus binary operator. *(number % 10)* returns the remainder when divided by 10. % may not be used with float or double operands.

'0' is the character zero in the machine's character set. In ASCII, '0' is represented by the number 48. As mentioned previously, *char* values in expressions are converted to *int*. Therefore *(number % 10) + '0'* returns an integer corresponding to one of the characters '0' through '9'—depending on the value of number.

As an exercise, write a version of *outdec* called *soutdec* that stores the digits as characters in an array rather than printing them. Also try writing the following useful functions.

```
isalpha (char)--return true if char is alphabetic else false
isdigit (char)--return true if char is numeric else false
islower (char)--return true if char is lowercase alpha else false
isupper (char)--return true if char is uppercase alpha else false
```

Thus far we have used only automatic variables. There are several other important variable classes and next time we will look at their use and scope. ▲

tradetalk

Δ As a result of an agreement with **IBM** (Boca Raton, FL), **Corona Data Systems** (Westlake Village, CA) has begun shipping its desktop and portable personal computers with a newly developed system ROM that allows the Corona personal computers to remain compatible with the IBM PC. A January stipulation, order, and judgment enjoined Corona from infringing any IBM copyrights; Corona agreed to the stipulation without admitting liability, and IBM agreed not to enforce the stipulation for computers shipped by Corona before mid-February. States Daniel R. Carter, Corona president, "We have been working our way through our extensive list of IBM PC-compatible software packages, and we have found that Corona has retained the same high level of compatibility." Δ **Rana Systems** (Chatsworth, CA) has purchased the worldwide manufacturing and marketing rights to Corona's line of plug-compatible Winchester disk drive subsystems, including those for the IBM PC-compatible add-on market.

Δ **VisiCorp** (San Jose, CA), licensed marketer of *VisiCalc*, a program created by **Software Arts** (Wellesley, MA), was denied a temporary restraining order and preliminary injunction enjoining Software Arts from using the trademarks "Visi," "VisiCalc," and "VisiCalc Advanced Version." Software Arts intends to proceed with plans to market *VisiCalc*, claiming that VisiCorp failed to maximize sales of *VisiCalc* as required by the terms of a 1979 agreement between the companies. VisiCorp recently announced that 700,000 *VisiCalc* packages have been sold since 1979. Δ The General Services Administration has awarded a federal supply schedule contract to Software Arts for 1984 for its equation-processing program *TK!Solver*.

Δ **Byte Industries**'s (Hayward, CA) wholesale distribution business will begin operating under the new name **Byte Distributes!** "The new name will help clarify that our wholesale distribution operation and our franchise activities have always operated separately," explained David Pava, president of Byte Industries. Byte Distributes! serves 2,500 accounts nationwide through its Hayward headquarters and represents forty manufacturers.

Δ After more than 50 percent of its employees donated blood when the Red Cross bloodmobile came by **Seequa**'s Odenton, Maryland, facility, the IBM-compatible manufacturer issued a challenge to other high-tech firms to meet or beat that percentage. "This is a good way for us to demonstrate that our industry is more than just microchips and disk drives," said Seequa president David Gardner. "This is our way of showing that we're human."

Δ **Albert Vezza** has been appointed chief executive officer of **Infocom** (Cambridge, MA). A founder of the four-and-a-half-year-old firm, Vezza will also continue to serve as chairman of the board. Vezza comes to Infocom from MIT, where he had been associate director of the Laboratory for Computer Science and where he led the Programming Technology Group, which developed a pioneering electronic message system; part of Arpanet, one of the first major computer networks; and a practical artificial intelligence system. Vezza's plans for Infocom? "We will strive to continue the rapid growth of the consumer products group, bring to market a new line of business software products that combines expressive power and new ideas about how computers should interact with users, and initiate development of new personal computer application areas."

Δ **C. Itoh Electronics** (Los Angeles, CA) announced that it has greatly increased its exportation of electronics products from the United States, primarily to Japan. President Mark Takeuchi said that the exports amounted to well over $100 million, consistent with CIE's goal of maintaining a zero balance of trade.

Δ **Ashton-Tate** (Culver City, CA) and **Reston Publishing Company** (Reston, VA) will codistribute selected titles from each company's line of computer publications. Ashton-Tate will distribute Reston publications supporting *dBase II*, including *User Friendly Guide to dBase II* and *Microcomputer-Based Business Systems: Featuring dBase II*. Reston will distribute Ashton-Tate's *Everyman's Database Primer* and *The Reference Encyclopedia for the IBM Personal Computer*. Ashton-Tate's newly created publishing group expects to produce more than thirty titles in the upcoming year. Δ The Ashton-Tate Publications Group has named **Cathy Bennett** sales and marketing manager. Bennett will be responsible for implementing marketing strategy and managing sales efforts for the group. Prior to her association with Ashton-Tate, Bennett served as marketing manager for Prentice-Hall, where she organized and managed a nationwide rep network, directing sales and promotion projects.

Δ **NewsNet** (Bryn Mawr, PA), the nation's largest electronic newsletter publishing service, premiered the controversial expose, "In IBM We Trust," in the *Annex Computer Report*, which monitors significant product,

price, and software development in the IBM and compatible marketplace. The story resulted from ACR editor **Bob Djurdjevic**'s four-month investigation of pending and recently resolved IBM court cases, including civil suits against the Gartner Group (Stamford, CT), Jonathan Fram (an ex-IBM employee currently working for the Gartner Group), and National Semiconductor/National Advanced Systems (Mountain View, CA), accused of obtaining IBM's proprietary information. At stake, says Djurdjevic, is the free flow of information within the computer industry.

Δ **Compaq Computer Corporation** (Houston, TX) reported sales of $112 million in its first full year of operation, resulting in net income of twenty-four cents per share.

Δ **Adventure International** (Longwood, FL) has signed a long-term licensing agreement with the **Marvel Entertainment Group** (New York, NY). Adventure International will create and market a series of graphic adventure games featuring the Hulk, Spider Man, and Captain America, while Marvel will release companion comic books called *Quest Probe* to tie into these games. Δ **Scott Adams, Inc.**, the parent company of Adventure International, has appointed **Greg Tibbetts** director of marketing for subsidiaries Adventure International and The Business Division. Tibbetts (not to be confused with Greg Tibbetts of Lobo Systems) is currently establishing an independent sales force.

Δ **Clarity Software**'s (Austin, TX) *3-2-1 Go*, which converts *1-2-3* worksheets into files containing **Execucom**'s IFPS modeling language (a set of English-like statements that document the spreadsheet's assumptions and calculations) can now be licensed to companies' foreign subsidiaries for an additional $1,000 per country. "Our marketing plan has been to license software to corporations under an organizationwide license for a single payment," Clarity president Jim DeLine said. "The corporation is permitted to make and distribute multiple copies of the software within their organization. Our customers save money by not being required to purchase more copies than they can use. The copy-protection problem is solved. Everyone seems pleased with this approach." A license fee of $2,000 permits corporate wide use of *3-2-1 Go* in the United States.

Δ **David A. Schmitt** has been named president of **Lattice Incorporated** (Glen Ellyn, IL), which designed the Lattice C compiler for the PC and compatibles. Schmitt comes to Lattice from Bell Telephone Laboratories, where he spent eighteen years as manager and engineer. He was recently in charge of the department responsible for the fault-tolerant version of the Unix operating system. Δ Also, **Francis L. Lynch**, principal author of the Lattice C compiler, has been named director of language products. "We will soon release compilers for the Z-80 and 68000," said Lynch, "and since they use the same front end and the same library as the 8086 compiler, software designers will be able to support all three chips easily."

Δ **Eagle Computer** (Los Gatos, CA) released its sales figures for the second quarter of fiscal 1984: $19,159,000, a 34 percent increase over sales for the previous quarter.

Δ The board of directors of **BPI Systems** (Austin, TX) has elected **Thomas W. O'Brien** vice president, product development. O'Brien will be responsible for development, quality assurance, maintenance, and enhancement of business accounting, the personal series, and productivity aids. O'Brien was previously senior vice president and manager of data processing at InterFirst Bank in Austin.

Δ **Schuchardt Software Systems** (San Rafael, CA), publisher of the *InteSoft* line of integrated business application software, has signed **Warehouse 1** (Canton, MA), a telemarketing firm, and **Software Wholesalers** (Avon, MA), a national distributor of business and development software, to national distribution contracts.

Δ **Business Solutions** (Kings Park, NY) plans to take on all existing integrated software product competitors to its *Jack2* integrated software. As part of the challenge, BSI is sponsoring a study by Harvard Business School students on the applications and benefits of integrated packages. Challenges will take place in seven major cities, and will consist of problem sets that each competitor will have to solve. Business school students will test the software while consultants from **Touche Ross & Company** (New York, NY) officiate.

Δ **Persyst** (Irvine, CA) has added seven organizations to its network of international distributors, as part of a campaign designed to establish a worldwide distribution and service network for the company's products by the end of 1984. The new distributors include Satt Electronics (Norway), Nordic Software AB (Sweden), Colruyt Informatics/Services (Belgium), Computer Assisted Management AG (Switzerland), Electrical Contracting Co. (Kuwait), and two South African firms: Grid Communicating Computer Center and Medusa Trading. Δ **Brad Freeburg** has been appointed international sales manager, responsible for the recruitment and management of international distributors. Before joining Persyst, Freeburg served as a sales representative for MSI Data Corporation and also managed its Latin American distribution system.

Δ **Computer Advanced Ideas** (Berkeley, CA) has changed its name to **Advanced Ideas** because the CAI acronym was trademarked by another firm. It has also recently moved to 2550 Ninth Street, Suite 104, Berkeley, CA 94710. Telephone: 415-526-9100. ▲

WATCH THE BIG GUYS

When you are going to expand a thousand IBM PC's you want to get it right. Fact is, if you are going to expand one PC you want to get it right. That's why world leaders in computers, energy, manufacturing, research and education as well as tens of thousands of individual computer owners have picked Qubie' for their PC or XT expansion products.

THE PEOPLES CHOICE

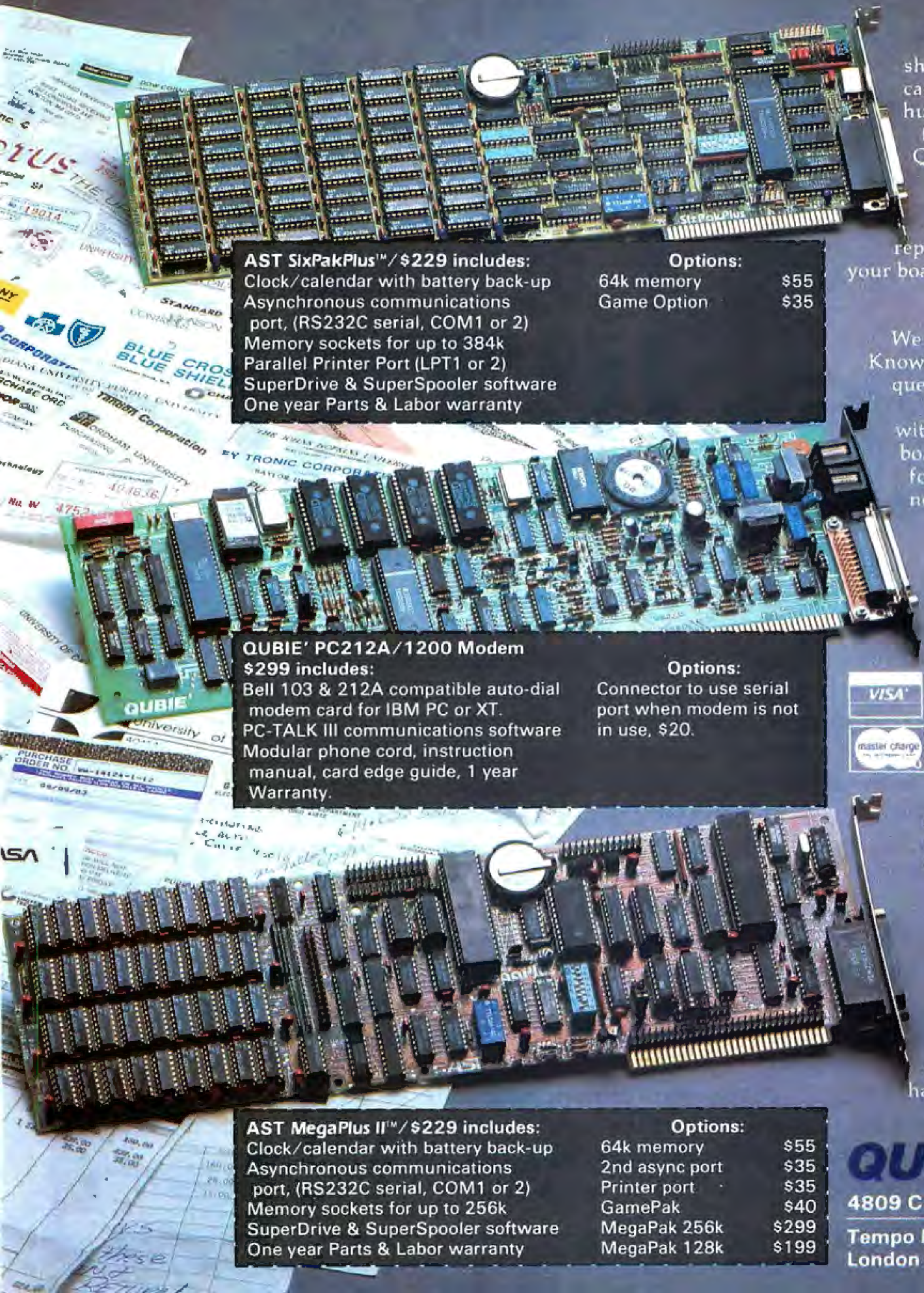
You can't gather an impressive list of customers without great products. You just have to be impressed with **MegaPlus II™**, **SixPakPlus™**, and the Qubie' **PC212A/1200** modem card. **MegaPlus II™** is the most flexible expansion product available for the PC. It's companion, the **SixPakPlus™**, has quickly become the number one selling multifunction board for IBM PC's. The Qubie' modem card utilizes four digital microprocessors to provide flawless performance at 300 or 1200 baud. It even comes with **PC-TALK III™** software, the program PC WORLD magazine called, "The benchmark that other PC communications packages are measured against". All this for under $300. No wonder the Fortune 500 is impressed!

THE RIGHT PRICE

Our prices don't just look good, they are. No extra charges for shipping, testing, insurance, credit cards, or COD fees. If you are in a hurry, 2 day air service is available for just $5 extra. We also offer a Qubie' exclusive, our PREFERRED CUSTOMER PLAN. For $50 we will extend your warranty a second year, perform warranty repairs within 24 hours, and return your board by UPS blue label air service.

DON'T FORGET SERVICE

We will be there when you need us. Knowledgeable people to answer your questions before and after you buy. Warranty repairs are performed within 48 hours or we replace your board. All products are guaranteed for one year on parts and labor. If not completely satisfied within 30 days of purchase, return your board for a complete refund including the postage. Join our list of impressive customers. You'll be glad you did.



AST SixPakPlus™/$229 includes:
- Clock/calendar with battery back-up
- Asynchronous communications port, (RS232C serial, COM1 or 2)
- Memory sockets for up to 384k
- Parallel Printer Port (LPT1 or 2)
- SuperDrive & SuperSpooler software
- One year Parts & Labor warranty

Options:	
64k memory	$55
Game Option	$35

QUBIE' PC212A/1200 Modem $299 includes:
- Bell 103 & 212A compatible auto-dial modem card for IBM PC or XT.
- PC-TALK III communications software
- Modular phone cord, instruction manual, card edge guide, 1 year Warranty.

Options:
Connector to use serial port when modem is not in use, $20.

AST MegaPlus II™/$229 includes:
- Clock/calendar with battery back-up
- Asynchronous communications port, (RS232C serial, COM1 or 2)
- Memory sockets for up to 256k
- SuperDrive & SuperSpooler software
- One year Parts & Labor warranty

Options:	
64k memory	$55
2nd async port	$35
Printer port	$35
GamePak	$40
MegaPak 256k	$299
MegaPak 128k	$199

TO ORDER BY PHONE:

In CA (805) 987-9741
Outside CA (800) 821-4479

VISA
master charge

TO ORDER BY MAIL:

Send a complete description of products, and your daytime phone number. Include check or credit card with expiration date. (Personal checks take 14 days to clear). California residents add 6% sales tax.

CORPORATIONS & INSTITUTIONS

Call for details on terms and delivery before mailing your purchase order. If your organization needs large quantities we have the prices and quick delivery you need.

QUBIE'

4809 Calle Alto, Camarillo, CA 93010

Tempo House, 15 Falcon Road, London SW11, United Kingdom

THE MOST POPULAR SOFT-WARE POLL

RESULTS!

The results are in, the votes have been counted, and the news is unambiguous: 1983 was the year that *1-2-3* demolished all opposition in its first appearance in *Softalk*'s annual software poll. How big did it win? Think Johnson—Goldwater. Think Nixon—McGovern. Think Chernenko. *1-2-3*'s final score was nearly three times greater than that of runner-up *WordStar*. *That's* a landslide.

First, a few words about how the Most Popular Software Poll results were tabulated. We recorded all the ballots we received on—what else?—a *1-2-3* worksheet, listing each software product in column A, counting up the number of first-place votes it received in column B, the number of second-place votes in column C, and so on. We then calculated a final score for each product by multiplying the figure in column B by 10, the figure in column C by 9, and so on across each row. The result was a weighted tally in which first-place votes counted ten times as much as tenth-place votes. Table 1 lists the thirty-five programs dearest to the hearts, minds, and fingertips of *Softalk/IBM* readers.

Although much of this year's most popular software appeared on last year's chart as well, half of this year's top ten finishers were making their first appearance. Other rising stars, besides *1-2-3* were *ProKey*, *Multimate*, *Word Perfect*, and *Word Proof*.

Illustration by Dan Winkler

TOP THIRTY-FIVE

		1	2	3	4	5	6	7	8	9	10	Votes	Score
1	1-2-3	271	86	44	28	24	7	8	6	4	5	483	4274
2	WordStar	46	52	52	27	18	14	17	7	2	6	241	1810
3	dBase II	39	44	28	26	15	15	10	11	6	7	201	1449
4	MS Flight Simulator	21	21	37	29	33	29	28	18	8	16	240	1439
5	Norton Utilities	7	14	44	33	42	34	26	13	20	16	249	1400
6	ProKey	34	33	23	20	14	16	16	8	7	8	179	1235
7	PC Talk III	17	25	34	24	12	9	4	7	6	3	141	1004
8	PC File	17	29	23	6	8	16	8	6	6	2	121	849
9	MultiMate	29	31	14	7	7	4	5	8	1	2	108	840
10	Volkswriter	20	24	22	5	8	2	5	5	4	1	96	729
11	Basic Compiler	11	15	13	17	16	17	10	5	4	8	116	720
12	VisiCalc	13	16	13	12	11	14	8	7	14	5	113	684
13	Multiplan	13	24	12	17	6	7	4	6	4	3	96	677
14	Word Perfect	31	19	6	6	4	1	5	3		2	77	631
15	Personal Editor	23	20	8	7	2	6	4	2	2	5	79	596
16	PFS:File	7	17	16	12	11	7	5	2	1	2	80	566
17	Word Proof	15	15	18	4	5	4	1	1	8	2	73	532
18	Macro Assembler	2	3	16	11	12	15	10	15	9	5	98	507
19	Smartcom II	4	13	9	13	8	6	5	7	1	2	68	443
20	Volkswriter Deluxe	19	15	7	3		5	1	2	1	1	54	440
21	Zork I	5	5	14	11	8	7	8	7	5	3	73	433
22	Crosstalk	5	11	13	9	9	4	4	4	4	4	67	430
23	SuperCalc2	9	16	5	5	6	4	5	4	1	2	57	401
24	EasyWriter 1.1	7	15	8	9	3	2	5	3	1	5	58	396
25	FriendlyWare PC Intro	2	3	8	6	10	14	9	10	3	6	71	361
26	Typing Tutor	1	6	4	5	16	8	8	9	4	9	70	343
27	Word	10	11	2	5	6	4	1	3	3	2	47	327
28	EasyWriter II	8	7	8	9	2	5	2	2		1	44	322
29	Olympic Decathlon	1	3	3	11	9	6	7	4	9	3	56	283
30	DOS 2.0	4	5	11	4	4	4	4	3	3	2	44	278
31	Home Accountant Plus	2	8	8	5	6	4	2	2	5	4	46	275
32	PFS:Write	10	7	4	5	4		2	1	2	1	36	270
33	Microsoft Adventure	2	2	6	7	6	3	6	5	4	5	46	238
34	PFS:Report		3	5	8	7	5	2	1	4		35	209
35	IBM Pascal	2	3	2	9	5	3	3	4	2	2	35	201

Some traditional heavyweights stood their ground. Examples include *dBase II*, *WordStar*, *Flight Simulator*, and the Basic Compiler. Others, like *VisiCalc* (last year's winner), slipped dramatically.

It's heartening to see that half of the top ten programs in the final standings were essentially single-person efforts, not corporate team productions. Peter Norton (*Norton Utilities*), David Rose (*ProKey*), Andrew Fluegelman (*PC Talk III*), Jim Button (*PC File*), and Camilo Wilson (*Volkswriter*) are evidence that as late as 1983 there was still room for the individual to make a mark in the PC arena. All five of these products succeeded without the benefit of megabuck marketing campaigns.

Another noteworthy point: Three of these individually developed programs fared much better in the popularity poll than they do in the monthly *Softalk* Top Thirty. The reason is clear; these products all have a significant component of nonretail sales, and our bestseller poll polls only retailers. Two of these five, Fluegelman's *PC Talk* and Button's *PC File*, of course, have essentially no retail sales.

Judging by this year's results, one would have to conclude that IBM didn't exactly consolidate its hold on the software market in 1983. Programs like the original *Adventure* (which plummeted from ninth place to thirty-third) faced stiff competition from more sophisticated games, especially those published by Infocom. *EasyWriter 1.1* dropped from tenth to twenty-second. *IBM Pascal* fell from nineteenth to thirty-fifth. Only the *Basic Compiler* and *Macro Assembler* maintained their strong showings of last year.

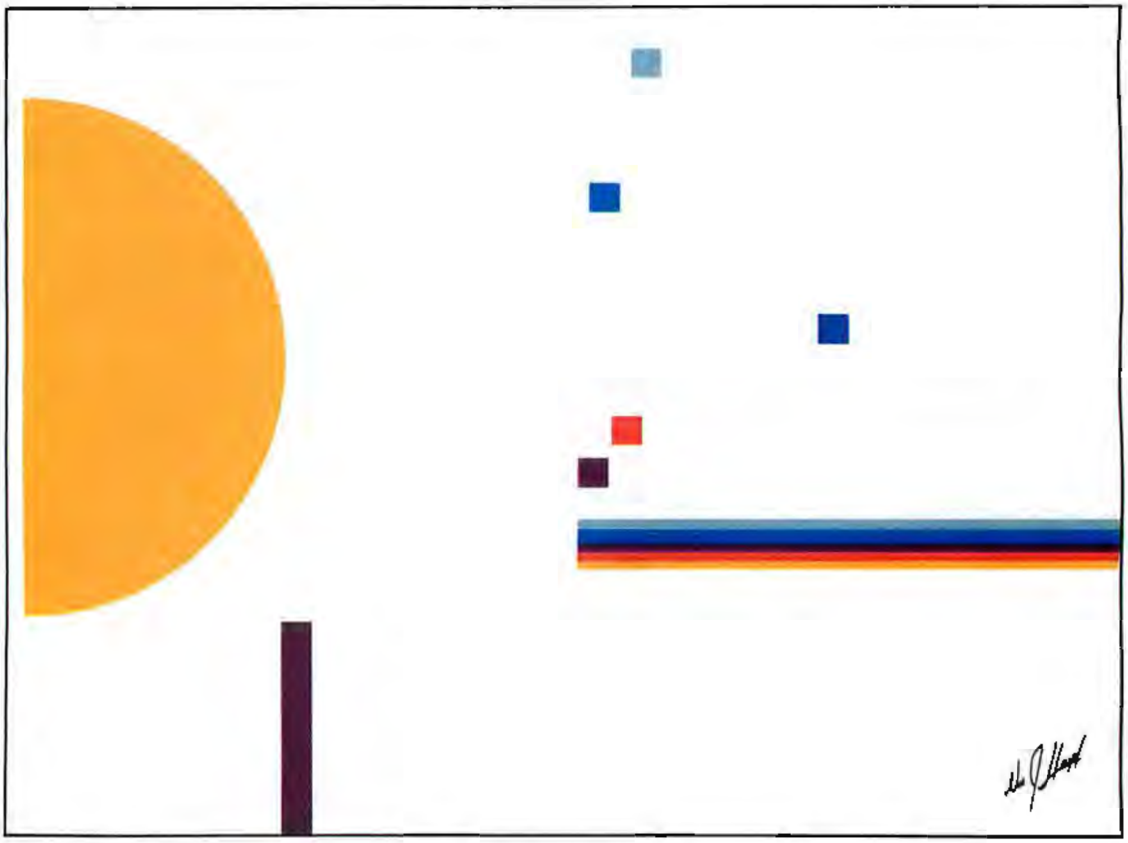
Table 2 ranks the top thirty-five in order of what we call their enthusiasm quotient (EQ). To measure the magnitude of the average vote a given product received, we divided each product's final score by the number of votes it received.

The term EQ may be a misnomer (though it's the best we could think of); it's reasonable to interpret this index as a measure not only of how much people like the various products they voted for but also of how often they use them. For example, the fact that the top of the list is *1-2-3* followed by a passel of word processors leads one to conclude that PC users do a lot of writing on their machines (*and* that they love *1-2-3*). That *EasyWriter 1.1.* is low on the list compared to the other word processors suggests that although this program has been around a long time, its users are something less than enamored of it. *Word Perfect*'s lofty ranking carries the opposite message: Since it has been on the market a relatively short time and (apparently) has yet to achieve optimum distribution, *Word Perfect* appears to be a hit with those who have it. The relatively low EQ of the *Norton Utilities*, one of the top ten programs in the overall standings, could well mean that many have it who only occasionally use it. This is no slur on Norton, because his utilities are not the sort intended for everyday use; compare the higher EQ of *ProKey*, a product designed for nearly constant use but one that finished slightly lower than Norton's in the aggregate point tally.

The stereotypic notion that the average PC user puts business before pleasure is borne out by the low EQ of the highest-positioned game in table 2—*Microsoft Flight Simulator*, which racked up an EQ of 6.00. Of course, this might just mean that many readers never landed successfully

On *1-2-3*....

I find *1-2-3* to be a very powerful package....*1-2-3* is way out front in spreadsheet software....*1-2-3*—excellent!....I love *1-2-3*, but it requires too many keystrokes. That's why I like *ProKey*....*1-2-3* is too bulky and requires too much disk and memory. I no longer use it ....Never found a bug in *1-2-3*....*1-2-3* is what software should be. Can't wait for next version ....Now that data management and on-screen graphics are offered by Sorcim, *SuperCalc3* is far superior to *1-2-3* in terms of overall usefulness....Lotus *1-2-3* and *ProKey* are outstanding....*1-2-3* outstanding business and statistic product....*1-2-3*, all show and no go....*1-2-3* is a new benchmark for software that you can use immediately and then grow into....*1-2-3* best rating.... We've done a tremendous amount of work with *1-2-3*; we couldn't have done the figure work to buy a company without it....*1-2-3* ∗ 10—there's nothing else!....*1-2-3* is all I need....*1-2-3*'s copy protection is inconvenient on my hard disk.



Position	Program	Votes	Score	Enthusiasm Quotient
1	1-2-3	483	4274	8.85
14	Word Perfect	77	631	8.19
20	Volkswriter Deluxe	54	440	8.15
9	MultiMate	108	840	7.78
15	Personal Editor	79	596	7.54
2	WordStar	241	1810	7.51
10	Volkswriter	96	720	7.50
32	PFS:Write	36	270	7.50
28	EasyWriter II	44	322	7.32
17	Word Proof	73	532	7.29
3	dBase II	201	1449	7.21
7	PC Talk III	141	1004	7.12
16	PFS:File	80	566	7.08
13	Multiplan	96	677	7.05
23	SuperCalc2	57	401	7.04
8	PC File	121	849	7.02
27	Word	47	327	6.96
6	ProKey	179	1235	6.90
24	EasyWriter 1.1	58	396	6.83
19	Smartcom II	68	443	6.51
22	Crosstalk	67	430	6.42
30	DOS 2.0	44	278	6.32
11	Basic Compiler	116	720	6.21
12	VisiCalc	113	694	6.14
4	MS Flight Simulator	240	1439	6.00
31	Home Accountant Plus	46	275	5.98
34	PFS:Report	35	209	5.97
21	Zork I	73	433	5.93
35	IBM Pascal	35	201	5.74
5	Norton Utilities	249	1400	5.62
29	Olympic Decathlon	56	293	5.23
18	Macro Assembler	98	507	5.17
33	Microsoft Adventure	46	238	5.17
25	FriendlyWare PC Intro	71	361	5.08
26	Typing Tutor	70	343	4.90

	COMMUNICATIONS	1	2	3	4	5	6	7	8	9	10	Votes	Score
1	PC Talk III	17	25	34	24	12	9	4	7	6	3	141	1004
2	Smartcom II	4	13	9	13	8	6	5	7	1	2	68	443
3	Crosstalk	5	11	13	9	9	4	4	4	4	4	67	430
4	Asynch. Comm. Supp.	1	2	4	2	4	3	1	1	2	3	23	127

	GAMES	1	2	3	4	5	6	7	8	9	10	Votes	Score
1	MS Flight Simulator	21	21	37	29	33	29	28	18	8	16	240	1439
2	Zork I	5	5	14	11	8	7	8	7	5	3	73	433
3	Olympic Decathlon	1	3	3	11	9	6	7	4	9	3	56	283
4	Microsoft Adventure	2	2	6	7	6	3	6	5	4	5	46	238
5	Frogger		1	2	10	6	7		5	7	1	39	196
6	FriendlyWare PC Arcade	1	3	3	5	8	3	2	6	4		35	193
7	Deadline			2	5	3	9	3	5	4	6	37	155
	Zork II	1	3	4	3	2	2	2	6	7	3	33	155
9	Night Mission Pinball	2	1	2	2	8	2	6	1	1	4	29	150
10	Zork III		2	5	3	2	2	4	6	2	7	33	146
11	Suspended	3	2	1	2	2	3	2	3	4	1	23	123
12	PC Man		2	5	1	1	1	3		5	7	25	105
13	Temple of Apshai	1		1	2	5	3	3		2	4	21	97
14	J-Bird				4	3	4	2		5		18	84
15	Adventure in Serenia	1		3		3	2	2	1		3	15	76
16	Starcross		1	1	3		2		1	1	2	11	55
17	Planetfall	1	1	1		2		2	1	2		10	54

THE MOST POPULAR SOFTWARE POLL

(maybe never got off the ground?) or figured out how to get past the axe-wielding troll.

Looking at breakouts of software categories, we see many cases where one program far outstripped its rivals. *1-2-3*, of course, completely overshadowed its competitors in the spreadsheet and data-handling categories. *WordStar* still, and by a comfortable margin, beat all comers in its department (notwithstanding the criticism that has been heaped upon it). *Microsoft Flight Simulator* soared over the other entertainment software. *PC Talk III* scored 600 points more than *Smartcom II*—not bad for a program in the public domain. *Word Proof* outdistanced *The Word Plus* in the word processing utilities section 2 to 1.

Of course, we'd be foolish to rely on statistics alone to learn how you feel about your software. It seemed like there was somebody with a good or a bad word for almost every one of the six-hundred-some products you voted for. Recurrent gripes were copy protection, obtuse documentation, and poor product support.

Our favorite comment came from the pragmatic soul who wondered; "Where *is* the rock without the scorpion in *Adventure in Serenia*!?" Anybody care to answer that one?

On WordStar....

WordStar, versatile, bug-free, and not copy-protected....Learning *WordStar* is as much fun as giving birth!....*WordStar* is still tops....Hate *WordStar* but use it....Ditched *SpellStar* for *Word Plus*—no regrets!....Paid good money for *WordStar*—regret it—ugh!....I hate *WordStar*....*WordStar*—for patchability....*WordStar* is the most overpriced, overrated program!....Used to like *WordStar*, must have been crazy!....*WordStar* is so much less capable than *Word Perfect* I can hardly believe it outsells it. *Word Perfect's* the best word processor....*WordStar* + *1-2-3* = software excellence....*WordStar* !@#$%&*....*WordStar* and PFS are superior....Give *WordStar* negative ten points for me....Only novices buy *WordStar*....*WordStar*—excellent!

CHICAGO
NEW YORK

FORMERLY APPLEFEST & PC'83

THE PERSONAL COMPUTER USERFEST

Plan now to attend the Personal Computer Userfests, the largest events ever . . . for Apple and IBM PC users.

Userfest brings together two of the largest, most successful shows ever conceived for personal computer users: Applefest and PC'83. Now that Apple and IBM can run each other's software, and with so many products adapted for both systems, the two shows merged beautifully.

At Userfest you'll see—and try out—all of the newest state-of-the-art products for your Apple, IBM PC or work-alike. Each Show has hundreds of displays and exhibits, and thousands and thousands of products including innovative new software, power peripherals, accessories, support services, books and publications. Products to help you explore the full potential of your computer for office, home and school applications.

Userfest features all the major makes of Apple and IBM computer compatibles. In fact, it's the largest display of these products, and biggest gathering of IBM and Apple experts, ever assembled in either city. Hence, you can learn more in two days at Userfest than you could in months of visiting computer stores or reading trade journals.

And best of all, everything on display at Userfest is for sale, usually at special show prices, so you can save hundreds, even thousands of dollars by making your purchases at the Show.

So don't miss the Personal Computer Userfest when it comes to Chicago and New York in 1984. It's a once-only opportunity.

Order your tickets in advance and avoid long lines. Admission is $10.00 for a one-day ticket, or $20.00 for four days. Children's tickets (under 10 years of age) are $4.00 and $8.00. If you need hotel accommodations and/or airline reservations, check the line on the Advance Ticket form.

FEATURING IBM, APPLE & WORK-ALIKE COMPUTERS & COMPATIBLES

THE PERSONAL COMPUTER

USERFEST

USERFEST/ CHICAGO

Thursday-Sunday
May 3-6, 1984
10:00AM to 5:00PM daily
O'Hare Exposition Center
9291 West Bryn Mawr
Rosemont, Illinois
(next to Chicago's O'Hare Airport)

USERFEST/ NEW YORK

Thursday-Sunday
September 20-23, 1984
Madison Square Garden
10:00AM to 5:00PM daily

For information about exhibiting at the Personal Computer Userfests, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000.

For hotel information call or write Trade Show Department, Fox Travel, P.O. Box 498, Waltham, Mass 02254. Tel: 617-890-1770 or 800-225-8410 ext. 314.

Userfest (formerly known as Applefest and PC'83) is produced by Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167.

ADVANCE TICKET ORDER FORM

Mail this form (or a facsimile) with full payment to Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000. No ticket orders accepted 14 days or nearer to each Show. Your tickets will be mailed one month prior to the Show. Sorry, no telephone or credit card orders please.

Name: ______________________________

Company (if any): ______________________________

Address: ______________________________

City: ______________ State: ________ Zip: ________

Tel: Day (____) ______________ Evening (____) ______________

Enclosed is full payment for

______ (quantity) adult one-day tickets @ $10.00 each ______ (quantity) adult four-day tickets @ $20.00 each

____

use this line for children's ticket orders

☐ Check here if you need hotel and/or airline information

THE MOST POPULAR SOFTWARE POLL

Point/Counterpoint

MultiMate 3.2 is much more powerful and useful (than *MultiMate 3.1*), but it's slower ....*MultiMate*, better than our dedicated word processor....*MultiMate* 3.2 solves earlier problems....The worst, dead last, lousiest of all = *MultiMate*! Line counts don't equal when format line is changed. Spelling checker can't handle hyphens....*MultiMate* is the ultimate word processor....*MultiMate 3.2* is slow, noisy, and too big....Most needlessly irritating software: PC-DOS 2.0/2.1....DOS 2.0 deserves honorable mention....DOS 2.0 isn't bad either....I like my DOS (by the way, my birthday is the 23rd)....*PFS:Write* is the best I have used....*PFS:Write* is so intuitive I feel like a computer pro without any computer training....*PFS:File* and *Report* are the greatest...All PFS programs are terrible...Hate Microsoft *Word*'s copy-protection scheme—very clumsy—otherwise *Word* would be second on my list....Compared to *MultiMate*, Microsoft *Word* is a toy for bored executives....Microsoft *Word*: best word processor for ease of use ....*Word* is excellent in virtual screen....Microsoft *Word* is collecting dust....I hate Microsoft's *Word*....Worst software: *Norton Utilities*....Norton is very helpful in troubleshooting....*Volkswriter Deluxe*—powerful and truly simple to use....Have most problems with *Volkswriter* printing....Have *Volkswriter Deluxe*, don't like it....Microsoft *Flight Simulator* is art!....*Flight Simulator* would have my vote if Microsoft had let me ctrl-alt-del out of the program....*Flight Simulator* is great, but it still has some bugs that cause it to lock up....Must power off PC to clear *Flight Simulator*—bad!....*Flight Simulator*—awesome program....*Flight Simulator* sure has lousy color....*dBase II*, no statistical capability-....Have but don't like *dBase II*, *SuperCalc*.... *dBase II*, an uncommonly good programming system....I write any programs I need in *dBase II*....*SuperCalc*s are by far the most user-friendly spreadsheet programs!....Regret *VisiCalc* purchase....*VisiCalc* would be tops if it had variable-width columns.... -1 for *VisiCalc* ....*Zork I*—best game ever for PC....The Zork games are some of the best....*Zork* is pointless and stupid....

	GENERAL-PURPOSE DATA HANDLERS	1	2	3	4	5	6	7	8	9	10	Votes	Score
1	1-2-3	271	86	44	28	24	7	8	6	4	5	483	4274
2	dBase II	39	44	28	26	15	15	10	11	6	7	201	1449
3	PC File	17	29	23	6	8	16	8	6	6	2	121	849
4	PFS:File	7	17	16	12	11	7	5	2	1	2	80	566
5	PFS:Report		3	5	8	7	5	2	1	4		35	209
6	Knowledge Man	2	3	1	8	3	1			2		20	138
7	Friday			7	2	3	1	1	2	10	2	28	125
8	RBase	2	5	1		1		1	2		1	13	90
9	InfoStar	2		1	2	2	2		3	2	1	15	78
10	Database Manager II	3	1	2	2				1	1		10	74
11	TIM III	1	1	3	2	1	1					9	68
12	VisiFile	1		1	2	2	2		1	1	1	11	60
13	Data Capture		1	2	2	1	2			1		9	57
14	TIM	1		3	2		1					7	53
15	MBA		3		1	1			1	2	4	12	51

	LANGUAGES AND OPERATING SYSTEMS	1	2	3	4	5	6	7	8	9	10	Votes	Score
1	Basic Compiler	11	15	13	17	16	17	10	5	4	8	116	720
2	Macro Assembler	2	3	16	11	12	15	10	15	9	5	98	507
3	DOS 2.0	4	5	11	4	4	4	4	3	3	2	44	278
4	IBM Pascal	2	3	2	9	5	3	3	4	2	2	35	201
5	Turbo Pascal	5	4	2	1	3	4	3	2	1	2	27	169
6	UCSD Pascal	9	3	1	1	4				4	3	25	167
7	BasicA	2	3		2	4	1	2	1			15	101
8	DesMet C Compiler	1	2	2	1	2	3	1	1	1	1	15	88
9	Microsoft Fortran	1	1	2	3		2	2	1		2	14	79
10	IBM APL		1	2	5	1	1	1	1			12	78
11	IBM Cobol	1		1	1	1	3	4		3	3	17	71
12	APL (STSC)	4	1			1	1	1				8	64
13	Lattice C	2		1	1	3			1			8	56
14	DR Logo	1	1	1	1	1	1	1		1	1	9	52

	SPREADSHEETS	1	2	3	4	5	6	7	8	9	10	Votes	Score
1	1-2-3	271	86	44	28	24	7	8	6	4	5	483	4274
2	VisiCalc	13	16	13	12	11	14	8	7	14	5	113	684
3	Multiplan	13	24	12	17	6	7	4	6	4	3	96	677
4	SuperCalc2	9	16	5	5	6	4	5	4	1	2	57	401
5	SuperCalc3	2	7	3	1			1		2		16	122
6	SuperCalc	2	2	4	2	1	2	3	1	1	1	19	118
7	MBA		3		1	1			1	2	4	12	51

	TUTORIALS	1	2	3	4	5	6	7	8	9	10	Votes	Score
1	FriendlyWare PC Intro	2	3	8	6	10	14	9	10	3	6	71	361
2	Typing Tutor	1	6	4	5	16	8	8	9	4	9	70	343
3	PC Tutor		2	2	5	1	4	3	2	2	5	26	122
4	Master Type			2	5	6	3		3		4	23	115
5	The Instructor			2	3	3	1	1	5		3	18	82

UTILITIES	1	2	3	4	5	6	7	8	9	10	Votes	Score
1 Norton Utilities	7	14	44	33	42	34	26	13	20	16	249	1400
2 ProKey	34	33	23	20	14	16	16	8	7	8	179	1235
3 Sideways		3	4	8	7	3	3	1	2		31	191
4 Copy II PC	2			8	7	5	5	6	3	3	39	190
5 File Command	5	6	2	2		3	5	3		1	27	179
6 Basic Dev. Sys.	6	2	4	2	1	3	1				19	149
7 JFormat		3	2		1	1					7	54

WORD PROCESSORS AND EDITORS	1	2	3	4	5	6	7	8	9	10	Votes	Score
1 WordStar	46	52	52	27	18	14	17	7	2	6	241	1810
2 MultiMate	29	31	14	7	7	4	5	8	1	2	108	840
3 Volkswriter	20	24	22	5	8	2	5	5	4	1	96	729
4 Word Perfect	31	19	6	6	4	1	5	3		2	77	631
5 Personal Editor	23	20	8	7	2	6	4	2	2	5	79	596
6 Volkswriter Deluxe	19	15	7	3		5	1	2	1	1	54	440
7 EasyWriter 1.1	7	15	8	9	3	2	5	3	1	5	58	396
8 Word	10	11	2	5	6	4	1	3	3	2	47	327
9 EasyWriter II	8	7	8	9	2	5	2	2		1	44	322
10 PFS:Write	10	7	4	5	4		2	1	2	1	36	270
11 Wordvision	10	4	2	2		2	2		3	1	26	191
12 Professional Editor	4	5	3	3	2	2	2	1	1	2	25	167
13 WordPlus-PC	3	5	5	1	1		1				16	132
14 Final Word	4	4	3	1		1		1		1	15	116
15 PeachText	2	2	2	5	1	1	2	1	1	1	18	114
16 PC-Write	4	1		1	3	1	2	3			15	96
17 Edix	3	3	1	2	2	1					12	96
18 Perfect Writer	2	2		2					1	1	8	55

WORD PROCESSING UTILITIES	1	2	3	4	5	6	7	8	9	10	Votes	Score
1 Word Proof	15	15	18	4	5	4	1	1	8	2	73	532
2 The Word Plus		1	3	3	2	1	2	1			13	82
3 Proofreader/Grammatik			2	3	4		2	2	1	1	15	78
4 Mailmerge		1	2	4			1	1		1	10	61
5 Fancy Fonts				2	2	2	3	1	1	2	13	55
6 Spellstar		3			2	1	1	2			9	54

Raves

ProKey is great!....With *STSC* I can whip up any database or word processor. It's so good that nothing else deserves mention....Two of my choices (*Pac-Gal/Smiley* and *Page*) are public domain software on bulletin boards....*Word Perfect* is the finest word processor on the market. Ideal for my law office ....Can't live without *File Command*. Perfect Software multiple screens are great! *DR Logo*—terrific, and excellent manual...*Crosstalk*, by far the best program I've seen....*Data Ease* is a wonderful database....*PC-File III* is great! Performance and price are unbeatable ....*ProKey* should get at least double or triple credit. By far the best ever....*Word Proof* is clean, beautiful, and great manual....Take a look at *XyWrite*—two guys in a garage have by far the best word processor....IBM's *Personal Editor* is extremely versatile and useful for $100....*PC-Talk III* is fantastic public-domain software....*PC Forth*!!!....I have a lot of programs I like....I never could spell (now I can—*Word Proof*)....Best of all are my own home-written systems....*Turbo Pascal* is the most exciting package I've seen in twenty years ....*Compiler Assist Program* is great; I no longer have to monitor my Pascal compiler ....More people like Jim Button (*PC-File*)!!!....*Scrnsave* and *Softalk* are priceless....*Delta Drawing* vote submitted by my six-year-old....*StatPlan* does it all for $49....No PC should be sold without *ProKey*....DeSmet C package is a real bargain at $109....Some of the very best software is under $50 these days....Best *Personal Finance* program is a life saver; very good....Excellent support from SSI on *Word Perfect*....

Gripes

Have very little experience so far. Biggest problem is instruction books, especially IBM. Tells "what" but not "how." Redundant in some things, void in other places....*Friday!*, slow and weak, but easy and fast to learn and use. Good for small files.....Did not like Friendly intro set....*VisiCalc*, rotten and unforgivable service...Purchasing software has much in common with buying a pig in a poke. Surprise surprise! Sometimes a dead pig. Or a sick pig....Worst-case list: *Home Accountant* and *WordStar*....Hate ads that use the word "call" instead of a price....*TIM III*—agonizingly slow, poor report formats....Won't buy programs I can't back up....*Dow Jones*, bad program, good data....IBM support has gone from bad to worse. It takes months to get answers to questions.... My Winchester hates copy protection ....Too many programs. I dream in little monochrome characters....Recent *ProKey* update is a drag; doesn't work on RAM disk ....Because of my RAM disk, I won't buy copy-protected software....IBM products are trash. All IBM documentation is useless ....Most PC owners are paying very high hardware and software prices to run software that's as bad as the old Apple or CP/M software or worse. This is very disappointing....Why hasn't there been more entertainment software of the quality of *Flight Simulator*?....Still waiting for some good strategy games....More freeware would help a lot....Boo to IBM Cobol....

The Printed Word

by John Dickinson

The Intelligent Printer, Part VI

Now that you can use Basic to tell your printer to deploy its intelligent features, it's time to return to theory and learn how to use printer features requiring variable data. These features give you substantial flexibility in such things as the vertical pitch (line height) your printer uses, the form (page) length, and the location of printer tab stops. With advanced models you can even use variables to set the distance between printed characters, specify the ribbon color used for printing, or select various international character sets.

The printer features you've seen before have been direct and do such things as change the density of character printing or take a single action, like issuing a form feed or beeping the printer's horn. These features need relatively simple printer command sequences, requiring only that you tell the printer which action to take or option to enable. Features that require variable data are a somewhat different matter—they require more complex command sequences using data you've supplied to set one or more internal parameters for your printer's computer.

These printer parameters affect the way simple printer commands work. For example, a command sequence that sets the printer's vertical pitch (line height) parameter affects the distance the paper travels when a line feed command is issued (and therefore sets the vertical line height); a command that sets form (page) length affects how much paper is fed through each time a form feed command is issued. Printer parameters usually have default values set when you switch your printer on. But the variable command sequences give you the opportunity to override the defaults and change how the simple commands work.

In previous installments of the Intelligent Printer series, you've learned that all printer command sequences are made up of ASCII characters and that the commands are most easily represented by ASCII character sequence numbers. While there are several formats in which these numbers can be represented, they all come out to the same thing.

You add the variable data to the command sequences used for setting internal printer parameters to indicate how you want the printer's parameters to be set. For example, using a command sequence that sets the printer's form-length parameter requires that you tell the printer the value you want it set to; the command wouldn't be much use (to you or your printer) unless you can pass this information along.

These *complex* command sequences are a bit more difficult to use than the direct commands, because you have to provide the data values for the printer parameters you want to set. Making things even more difficult is the variety of syntax and numeric formats used by printer designers for transmitting the variable data to the printer. Some printers even use more than one format. The final results always transmit as ASCII characters, but it may not be a simple matter to figure out what the correct characters are for a complex command sequence.

Adding more complexity to your life may not seem to be worth it; many printers have simple command sequences available that perform a subset of the complex commands, and it's often tempting to get by with them. Providing the data for complex command sequences requires you to learn even more rules about your printer's programming language and, depending on the model, more about numbering techniques. But you'll gain a lot of flexibility from your printer by taking a little time here to learn how to use the more complicated commands. And, like printer programming rules you've learned before, the new ones are simple to use once you understand them.

Let's start by going over the complex command sequences available for our example printer (the Epson MX-80 without Graftrax), and look at a couple of simple command sequences that can do some of the same tricks. Your Basic printing knowledge will get a little workout when we do some exercises that show how these new commands (and their simpler companions) work. Later on you'll also look at some of the alternative data formats used by other printers for complex command sequences.

Name	Function	Decimal ASCII code
SPCN72	Set line feed to N/72nds of an inch	027 065 N
Formline	Set forms to N Lines	027 067 N
SetVT	Set vertical tabs at N1,N2 . . . etc.	027 066 N1 N2 . . . etc.
SetHT	Set horizontal tabs at N1,N2 . . . etc.	027 068 N1 N2 . . . etc.

Table 1. Epson MX-80 complex command sequences.

The MX-80 uses complex command sequences for variable line pitch (vertical line height), for variable form length, and for setting the printer's vertical and horizontal tab stops. The four complex command sequences (your printer may not have all of them, or it may have a few more) are listed in table 1. On the MX-80, you can also set vertical pitch to alternate fixed heights by using simple command sequences.

We're not abandoning the functional area organization of features used to develop our printer reference card, but, since these complex command sequences require additional information in order to be used correctly, let's extract them from the MX-80's reference card and treat them separately. When we've found out everything we need to know about these command sequences and put the information into a table, we can append it to our MX-80 reference card. You may want to do the same thing with the reference card you've made for your printer.

Each *N* in the complex command sequences in table 1 represents the required variable data value. The value of *N* used will set the printer's parameter; its position indicates where the MX-80 expects the data to be in the command sequence. Where *N* is followed by a number (for example, *N2*), more than one variable value may be used to set multi-

Missing an installment of "The Printed Word"? All back issues of the column—from August 1983—are still available; for further information, see page 4.

"MAYBE IT WILL MAKE A NICE PLANTER."

Is your home computer doing what you thought it would when you bought it?

If not, you should know about the entirely *new* personal finance management program from Sundex Software. Before, you guessed about your financial situation. With Certified Personal Accountant™ (CPA), you *know* where your money is and where it's going. In less time than it now takes to pay your bills by hand, you can do it on your personal computer. So whenever you want, you can be automatically updated on your tax position, net worth, budgeted expenses and cash flow.

We talked to hundreds of people who use the current best-selling personal finance programs. They told us what they *really* wanted. And we developed CPA. Here's what makes CPA different:

- There are 120 pages of instructions built right into the program that are ready to be displayed when you need a little help.
- CPA is ready to use immediately. There is no set up as required in other programs.
- Unlike other programs, you can add, delete, change and classify all transactions from the same screen.
- Every transaction can be assigned to both budget and tax categories.
- CPA is faster because it's not written in BASIC.
- Any CPA report can be displayed on screen or printed.
- Only CPA will feed information into our tax preparation and investment packages.

Don't take our word for it. Find out for yourself. Try the CPA at your local computer store. Or, call us toll-free at 800-835-3243 for the store nearest you. And while you're there, ask about our other Sundex products, the Certified Personal Investor™ and Personal Payables™. Any one will make your computer do what it's supposed to do.

Sundex

Sundex Software Corporation
4755 Walnut St., Boulder, CO 80301 303/440-3600

Available for: IBM® PC and PC jr • Apple® II and IIe • Commodore 64 • TI Pro • Franklin • Compaq • Corona • Eagle® 1600 • Columbia • Hyperion

ple parameters. All complex command sequences require at least one data value.

Table 2 shows the printer parameter set by *N* for each command sequence, the simple command whose action is affected by the variable value, and the default value in effect for the parameter at the time the MX-80 is switched on.

Your own printer may be somewhat different from the MX-80 (even if it's another Epson or a compatible printer). For example, it may have a different complex command sequence to set the same type of parameter, and the parameter itself may not be exactly like the MX-80's. One common difference is for printers to set the vertical pitch in forty-eighths, one hundred forty-fourths, or two hundred sixteenths of an inch, rather than the MX-80's seventy-seconds. Some printers even have more than one way of setting this parameter; Graftrax-equipped MX-80s can set vertical pitch in seventy-seconds or two hundred sixteenths of an inch and can set the form length in inches as well as lines. Other differences are equally common. Some Epson printers don't have vertical tab stops, and the RX-80 uses a different method for setting horizontal tabs. It's also possible that your printer has no equivalent complex command sequences.

Name	Parameter set by N	Affected Command	Default Value
SPCN72	Number of 72nds of an inch per vertical space	LINEFEED	12
FORMLINE	Number of vertical spaces per form (page)	FORMFEED	66
SETVT	Number of vertical spaces to each vertical tab stop	VT	none
SETHT	Column (space) position for each horizontal tab stop	HT	8,16,24. . .

Table 2. Epson MX-80 complex command sequence printer parameters.

As with parameters for most computer programs, printer parameters have limits that determine the values that will work in complex command sequences; these limits in turn are controlled by the design of your printer's hardware and the program running the printer. Each parameter usually has maximum and minimum limits, and command sequences that set more than one parameter (tab stops, for example) are also limited in the number of values that can be entered. Complex command sequences always require at least one variable value to be entered.

Printer parameter limits are similar to limits found in spreadsheet or word processing programs. Applications such as these limit the number of columns and rows (lines, pages, formulas, rulers, or whatever) you may have in a spreadsheet or document. In most applications where multiple parameter values can be set, at least one value must be provided.

Printer parameter limits are usually documented in the printer manuals, but they're often difficult to find and are occasionally left out. If you can't find limit specifications in your manual, your only recourse will be to experiment until you figure out the parameter limits. You'll know when you've reached a limit: Your printer won't respond correctly, or it will simply crash and stop functioning.

The MX-80's user manual documents the commands' variable limits; these are displayed in table 3, along with the rest of the information we have about these commands. You should try to find the limits for your printer's complex commands in its manual and add them to your table. Be cautious, however; if you have an Epson printer other than the MX-80, its limits are likely to be different.

Now you know which MX-80 command sequences are complex. You have all the information about the parameters they set, limits that should be observed when using the commands, simple commands that are affected by the parameters, and the default values that the parameters have when the printer is switched on. (Remember, the default values determine how a command behaves if you don't use the complex commands.) That's everything you need to know about the MX-80's complex command sequences—except how to use them!

Name	Variable set by N	Min N	Max N	Max #	Affected command	Default value
SPCN72	Number of 72nds of an inch per vertical space	1	85	NA	LINEFEED	12
FORMLINE	Number of vertical spaces per form (page)	1	85	NA	FORMFEED	66
SETVT	Number of vertical spaces to each vertical tab stop	1	255	64	VT	none
SETHT	Column (space) position for each horizontal tab stop	1	127	112	HT	8,16,24. . .

Table 3. Epson MX-80 complex command sequence printer parameters and limits.

Just as command sequences vary among printer makes and models, methods for transmitting variable data information in complex command sequences also vary. There are three basic methods commonly used by printer designers, plus a variation or two. You'll need to consult your manual to find the exact method used by your printer.

Before you look in your printer's manual for its variable data format, however, you might find it helpful to review the February installment of this column, which discussed the decimal, hexadecimal, and character number formats manuals use for printer command sequences. Manuals generally (but not always) use the same number format they employed to describe command sequences to indicate how complex command sequences transmit variable data. We'll give you a hand with this project next month.

Epson printers, and their compatible cousins and friends, have the easiest method for transmitting data in complex command sequences, so let's start there. Next month we'll provide a table of the other common methods, along with a couple of examples of how you can use each one and some BASIC program exercises to help you try them out.

In Epson-compatible printers the ASCII sequence number of the character used in the variable (*N*) position of the complex command sequence determines the value of the printer parameter set by the complex command. That means that if you want to transmit a data value of 24, all you have to do is send ASCII character number 24 in the *N* position of the complex command sequence. An example or two will illustrate how this works.

Suppose you want to set the vertical pitch (line height) to print double-spaced. Single spacing is normally six lines per inch, making each line one-sixth of an inch high. Vertical spaces on the MX-80 are defined in seventy-seconds of an inch, and one-sixth of an inch translates to twelve seventy-seconds. (Table 3 shows that twelve is the MX-80's default setting for SPCN72.) Double spacing is twice as high—or twenty-four seventy-seconds (one-third) of an inch. (You might find it useful to keep a hand calculator around to work out these fractions.)

The MX-80's SPCN72 command sequence can easily change the line height to twenty-four seventy-seconds of an inch. All you have to do is substitute an ASCII character 24 in place of *N* in the command sequence for SPCN72, like this:

```
027 065 024
```

This command sequence will work correctly to set up the MX-80 for double-spaced printing. Note that the value for *N* falls within the limits set by the printer's designers, and that one value (no more, no less) is provided.

Owners of MX-80-compatible printers (and other Epson models) should be careful with limits and other minor differences in the complex command sequences. Some printers use different maximums than the MX-80, and some use a slightly different command sequence to achieve the same result. (See the comparison of the IBM Graphics and Epson MX-80 printers in the September 1983 "Printed Word" for details about differences in one compatible model.)

To return the MX-80 to normal (single-spaced) printing, you have two choices. The first is to exercise the complex SPCN72 command sequence, this time using a parameter value of 12 instead of 24:

```
027 065 012
```

If you check your reference card for the MX-80 (or any other printer), you'll notice that ASCII character 12 is normally used for a form feed command sequence. In this case, however, the ASCII 12 in the command sequence will not be interpreted as a form feed command, because it's used in a complex command's variable position and not as a command sequence by itself. Any ASCII character in the complex command sequence's variable position will be treated as a data value instead of a printable character or command sequence. The ASCII 12 will be used, instead, to set the vertical pitch back to single spacing (twelve seventy-seconds or one-sixth of an inch).

The other way to return the MX-80 to single spacing is to use the simple command sequence designed for that purpose. If you check the MX-80 reference card, you'll see that the command sequence for SPC6LPI is:

```
027 050
```

There is an equivalent simple command—027 048—for setting the line spacing to one-eighth of an inch (SPC8LPI), but you can achieve the same result by using SPCN72 with a data value of 009.

Setting the form (page) length on the MX-80 is just as easy as setting the line height. The FORMLINE command sequence has the same data format as SPCN72 but sets the page-length parameter instead of

the line-height parameter. The default setting for the page length on an MX-80 (and most other printers) is sixty-six lines per page, which is the length of standard letter-sized (eleven-inch) paper. If you want to use legal-sized (fourteen-inch) paper and have the form feed command sequence operate correctly, all you have to do is set the form length to eighty-four (six times fourteen) lines per page. The FORMLINE sequence for this change is:

```
027 067 084
```

After you enter this command sequence, the form feed command sequence will feed a legal-sized page through your printer.

Another common reason for overriding the default form length setting is to produce mailing labels. Most labels are one inch high, so you need to set the printer's form length for six lines. This version of the FORMLINE command sequence should do the trick (use a different data value for longer or shorter labels):

```
027 067 006
```

Whatever the height of your labels, you'll have to use the FORMLINE command sequence to return things to normal (sixty-six lines per page) because there's no simple command sequence to do it for you:

```
027 067 066
```

Setting horizontal and vertical tab stops on the MX-80 poses a somewhat different problem. The SETHT and SETVT commands set the column or line positions for horizontal (HT) and vertical (VT) tab command sequences. Because these command sequences can set more than one tab stop (but not more than the maximum listed in table 3), you have to indicate how many you are setting whenever you use the commands.

On the MX-80 a terminator value, or delimiter, is used to tell the printer when you've finished setting tab stops. When it receives the terminator, the MX-80 knows that you're through entering values and processes the next character received as a new printable character or command sequence. If you don't tell the printer when you're through, it will interpret all incoming characters as tab-stop settings until it has received the maximum number of settings. The MX-80 and all Epson-compatible printers use an ASCII character 0 (NUL) as a terminator.

Use the tab-stop values you want to set to determine the ASCII characters for the rest of the SETHT command, just as you determined values for the SPCN72 and FORMLINE command sequences; be aware, however, of an additional rule. Values for SETHT must be entered in ascending order, because the MX-80's computer processes them sequentially, just as you would do when setting tab stops on a typewriter. The MX-80 will simply ignore values that are out of ascending sequence. You'll do no damage to most printers if you disregard this rule, but you could get unexpected results when you use the HT (horizontal tab) command.

The MX-80 also uses tab stops sequentially. Each time you issue an HT command, the print head moves to the next tab stop to the right of its current column position. The tab-stop sequence is refreshed each time you issue a carriage return. Both tab-stop characteristics mimic the action of a typewriter.

The MX-80 has default horizontal tab stops set at every eighth position, in compliance with ASCII file format and data transmission standards. The SETHT command sequence can be used to override these standard settings—to set tab stops, for example, at every five columns through character position thirty. The SETHT sequence to do this is:

```
027 068 005 010 015 020 025 030 000
```

Note that each column where you want a tab stop is represented by its ASCII character number, just as other data values are represented for Epson printers; and the entire sequence is terminated with an ASCII 0.

After you have entered this version of SETHT, all tab stops to the right of column 30 will be cleared. If you issue an HT command when the print head is beyond column 30, the command will operate incorrectly and cause a blank space to be printed.

Clearing the horizontal tab stops is simple—just tell the MX-80 that there are no tab settings by entering the following:

```
027 068 000
```

The ASCII 0 terminator following no values tells the MX-80 to forget about horizontal tabs (in which case HT will just print a space). If you want to reset the tabs to the ASCII standard, you have to reenter all the correct column values using the SETHT command sequence. (You can also reset them by switching the printer off and then on, but in the process you'll lose any other printer settings you may have established.)

We should make one other point about horizontal tab stops before moving on. The character font (width) in effect at the time tab stops are used (with the HT command sequence) determines exactly how far the head moves for each stop. The stops are set in character column positions (just as they are on a typewriter), and the head will not move as far in compressed print mode as it will in normal or wide mode. It will, however, move the same number of character positions in all cases.

The SETVT command sequence sets vertical tab stops in the same way SETHT sets horizontal stops. In this case, you're telling the MX-80 which vertical line to skip to each time you issue a VT (vertical tab) command sequence. Vertical tabs, like horizontal tabs, work in sequence and must be entered in ascending order.

If you're printing a form requiring output only on lines three, seventeen, and fifty, you can use the SETVT command to set vertical tab stops and use the VT command sequence to reach the line you need to print on. This strategy will improve printing speed, since (on most printers) vertical tab stops operate more quickly than an equivalent number of line feed commands. To set the desired vertical tabs, issue the following version of SETVT:

```
027 066 003 017 050 000
```

No tab stops will be in effect after line 50, and if you issue a VT at that line or beyond, it will operate as a line feed. The distance traveled when you issue a VT command is determined by the vertical pitch in effect at the time, just as the distance traveled in response to an HT command is governed by the character width in effect.

Clearing vertical tab stops is just like clearing horizontal ones. Just use the SETVT command sequence with only a terminator value:

```
027 066 000
```

When you use the VT command, you should remember that the vertical tab sequence is refreshed each time a form feed command sequence is issued (just as the horizontal tab sequence is refreshed with each carriage return).

You were promised a little Basic exercise for trying out the MX-80's complex command sequences, so here it is. We'll get you started, and you can let your imagination take you from there.

First, here's a fun way to look at variable line spacing. We'll use a Basic *for* loop to see the effect on the printer of all possible values for the MX-80's SPCN72 command (see listing 1). This exercise is a good one, because it demonstrates how to use a *for* loop variable (or any other kind) as a printer command sequence variable—something you may want to do more often than you might at first think.

Each time the program goes through the *for* loop (starting on line 1000) it redefines SPCN72 to set a new parameter, the value of the loop counter (SPC.N72NDS). It also prints out a line each time, and you'll see the vertical spacing get smaller and smaller as the *for* loop continues

to decrease the value of SPC.N72NDS. You can change this program to see what happens to the printer when you exceed one of the limits for the SPCN72 command sequence.

You should also try to write a variation of the program that works with the variable page length (FORMLINE) command. Make sure you have plenty of paper available before you run this program or any other program you use to experiment with the FORMLINE and form feed commands!

Another good BASIC exercise is to work with the printer's tab stops. Not many software packages take advantage of printer tab stops (horizontal or vertical), but you can do it in your own BASIC programs and get better printing speed as a result. Listing 2 has a simple program for setting the horizontal tab stops at 5, 10, 15, and so on.

You can run this program as is to see how these short stops work, or you can change the values (modify line 2010). But you can learn more from the program than that. For example, try the program with different character fonts to see what happens. Or use variables to set the stops. For example, substitute the *for* loop shown in listing 3 for line 2010.

```
1000 FOR SPC.72NDS% = 85 TO 1 STEP -1
1010    SPCN72$ = CHR$(27) + CHR$(65) + CHR$(SPC.72NDS%)
1020    LPRINT SPCN72$ + "The next LINEFEED is at: ";
1030    LPRINT USING "##"; SPC.72NDS%;
1040    LPRINT "/72nds of an inch . . ."
1050 NEXT SPC.72NDS%
```

Listing 1.

Another good exercise is to write the same kind of BASIC program for vertical tab stops, and work with SPCN72 to see the effects of different line heights. Once again, be sure you have lots of paper loaded when working with vertical feeds of any kind!

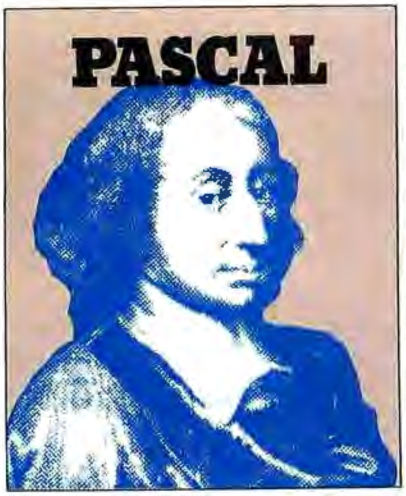
Next month we'll cover the data formats used by other makes and models of printers and we'll present some BASIC programs that work with them. Most of these commands are more difficult to use than the ones used by Epson printers, so if you have another make, be sure to tune in. ▲

```
2000 HT$ = CHR$(9)
2010 SETHT$ = CHR$(27) + CHR$(68) + CHR$(5) + CHR$(10) +
     CHR$(15) + CHR$(20) + CHR$(25) + CHR$(30) + CHR$(0)
2020 LPRINT SETHT$;
2030 FOR TAB.STOP% = 1 TO 5
2040    LPRINT HT$; "!<";
2050    LPRINT USING "##"; TAB.STOP% * 5;
2060 NEXT TAB.STOP%
2070 LPRINT " . . . Horizontal tabs set as indicated"
```

Listing 2.

```
2010 SETHT$ = CHR$(27) + CHR$(68)
2012 FOR TAB.STOP% = 5 TO 30 STEP 5
2014    SETHT$ = SETHT$ + CHR$(TAB.STOP%)
2016 NEXT TAB.STOP%
```

Listing 3.



FROM BEGIN TO END

by Bruce Webster and Deirdre Wendt

When Niklaus Wirth designed Pascal, he did so in a punched card/mag tape/mainframe environment, where fixed-length data was the rule. At least, that appears to have been the reason he was satisfied to store a character string as a *PACKED ARRAY[1..n] OF char.* Since almost every Pascal implementation has an explicit *string* data type, his decision may have been a little shortsighted. Or maybe not. At any rate, Standard Pascal does not (currently) have a predefined data type for strings. String constants such as

```
CONST
   filename      = 'B:STARS.DAT';
   lifename      = 'WHATEVER YOU WANT';
```

are considered to be *PACKED ARRAY[1..n] OF char,* where *n* equals the number of characters in the string.

The closest to a standard string data type in Pascal is that defined by UCSD Pascal. Not only is it used by all versions of UCSD Pascal (include those on the IBM PC), but it is also followed by Pascal/MT+, Turbo Pascal, and other Pascal implementations, and a limited version is supported in the utility library for Modula-2. The major exception is IBM Pascal, which has its own versions (two of them) of strings. Dara Pearlman did an admirable job of explaining them in the April 1983 issue of *Softalk* ("Pascal Extended," Part I) so we'll limit our discussion to the UCSD standard.

Why Strings? The early mainframe (read very, very large) computers were *batch-oriented* systems. "Jobs" were "submitted" by means of large decks of punched cards or reels of magnetic tapes, and results came out on high-speed line printers. But the arrivals of interactive operating systems ("timesharing") and the CRT terminal started a new approach to computer use. When the minicomputer showed up, so did the first code designed for interactive manipulation of text: word processing programs. The explosive growth of the microcomputer market over the last ten years has been matched by an equal growth in word processors and the number of people using them. For all its reputation as a number-crunching machine, the computer is used most often to move words, not values.

The basic concept behind text manipulation is the *string*. A string is simply a list, of some specified length, of characters. For our purposes, the characters belong to the ASCII character set and may be letters, digits, or punctuation. The string can even include nonprinting (control) or special characters. We can extract parts of the string, add characters to or subtract them from the string, combine the string with other strings, print the string out, read it in—in short, we can manipulate it.

Missing an installment of "Pascal B to E"? All back issues of this column—from July 1982—are still available; for further information, see page 4. The columns will also be published soon, as a single volume, by Softalk Books.

The UCSD String Standard

The UCSD Standard. UCSD Pascal (with Pascal/MT+, Turbo Pascal, and others) allows us to declare a variable to be of type *string,* with an optional length specification. For example, we can define the following:

```
VAR
   my_name       : string;
   token         : string[15];
   big_string    : string[255];
```

The default length is 80, so the variable *my_name* can hold as many as eighty characters. The variable *token* can hold only fifteen characters, so the statement

```
token := 'this is too long a string for token';
```

would cause a runtime error ("STRING OVERFLOW"). If you're really unlucky, this statement might not give you an error at all, instead overwriting the variables and/or code located near *token* and thereby causing all kinds of problems. The third variable above, *big_string,* represents the maximum length possible for a string.

The data type *string[n]* can be thought of as a *PACKED ARRAY[0..n] OF char.* For example, we can reference individual characters in *token* using the notation *token[1], token[2],* and so on. The first location, *token[0],* contains the current length of *token.* For example, if we execute the statement

```
token := 'this string';
```

then *token[0]* contains the value 11, since there are eleven characters in *'this string'*. However, we could *not* do something like this:

```
VAR
   token         : string[15];
   len           : integer;
BEGIN
   token := 'this string';
   len := token[0];
   writeln('The length of token is ',len)
END;
```

Why not? Because *token[0]* is of type *char,* and we can't assign a character to an integer. We could, however, substitute the statement

```
len := ord(token[0]);
```

which would return the ordinal value of *token,* which happens to be 11.

Using the array notation, we can play with any individual character of a string. As mentioned above, each element of a string is a variable of type *char* and can be treated as such. For example, we might want a procedure to convert all letters in a string to uppercase ('A'. .'Z'):

```
PROCEDURE low_to_up(VAR str : string);
VAR
   indx,len        : integer;
```

```
BEGIN
   len := length(str);    { more on this later }
   FOR indx := 1 TO len DO
    IF str[indx] IN ['a'. .'z']
      THEN str[indx] := chr(ord(str[indx])-32)
END; { of PROC low_to_up }
```

A few caveats are in order. Unless you know what you're doing, you should avoid messing with any elements beyond the current length of the string. In some Pascals, the statement *str[indx] := ⟨some char value⟩* when *indx ⟩ length(str)* causes a runtime error; and if it doesn't, you've just changed a portion of the string that won't print out unless you change the length as well.

We've already seen that we can assign string constants to string variables (*token := 'this string'*). We can also read and write strings through text files (see the September and October 1983 installments of this column). Most notably, we can read and write strings through the predefined text files, *input* and *output*. For example, the following program allows you to type in a line (up to eighty characters!) and then writes it back out to the screen. It continues to do this until you type, "I quit!":

```
PROGRAM echo;
VAR
   line              : string;
BEGIN
   writeln('Entering echo mode — type "I quit!" to exit');
   REPEAT
      readln(line);
      writeln(line)
   UNTIL line = 'I quit!'
END. { of PROGRAM echo }
```

Let's make the following modifications:

```
PROGRAM micro_word;
VAR
   line              : string;
   outfile           : text;
BEGIN
   write('Enter  filename: '); readln(line);
   rewrite(outfile,line); { or however you open a file }
   writeln('Entering insert mode--type "I quit!" to exit');
   REPEAT
      readln(line);
      IF line ⟨⟩ 'I quit!'
        THEN writeln(outfile,line)
   UNTIL line = 'I quit!';
   close(outfile,lock); { or however you close a file }
END. { of PROGRAM micro_word }
```

Voila! You've just written a word processor. However complex or sophisticated they may seem, all word processing programs eventually boil down to the program above. Start with this program, add modifications, and eventually you'll have your own text editor.

String Comparisons. Just like numbers, strings can be compared to each other. In our program *micro_word*, we checked to see whether the string variable *line* was equal to the string constant *'I quit!'*. The comparison was simple. First, the lengths of the two strings were compared. Differences in length meant that the strings were not equal. If the lengths were the same, the characters in the two strings were compared, starting with the first one and continuing until either (1) two characters were found to differ or (2) all characters had been compared. In case (1), the strings were not equal; in case (2), they were.

We could even write a function to show this comparison:

```
FUNCTION equal(str1,str2 : string) : Boolean;
{
        purpose    show how strings are compared for equality
                              Note well:This function is *not* necessary
}
VAR
   len,indx          : integer;
   flag              : Boolean;
BEGIN
   equal := false;
   len := ord(str1[0]);
   IF len = ord(str2[0]) THEN BEGIN
     indx := 1;
     flag := true;
     WHILE flag AND (indx ⟨= len) DO
         IF str1[indx] = str2[indx]
           THEN indx := indx + 1
           ELSE flag := false;
     equal := flag
   END
END; { of FUNC equal }
```

Once again, we'd like to stress that you do not need this function. We're just showing how the Boolean expression *str1 = str2* comes up with a value of *true* or *false*.

Similar comparisons occur when we want to see if one string is "greater than" or "less than" another. For example, let's suppose we're sorting a list of names into alphabetical order. At some point, we'll compare two strings to find which comes before (is less than) the other. The statement

```
IF str1 ⟩ str2 THEN . . .
```

takes some action if and only if *str2* comes before *str1*. The comparison algorithm can be described as follows:

(1) Point to the first character of each string
(2) Compare the two characters
(3) If they are not the same, go to (8)
(4) Get the next two characters
(5) If they're both there, go to (2)
(6) If only one string has characters left, then it is greater than the other
(7) Otherwise, neither string has characters left, so neither is greater than the other (they're identical!). Stop
(8) One character has a greater ASCII value than the other. The string that character came from is greater than the other string. Stop.

If you read the review of Turbo Pascal that appeared in this column last month, you may remember that it mentioned a bug in the string comparison code. Step (6) appears to be skipped altogether, so that the expressions *'abc' ⟨ 'abcde'* and *'abcde' ⟨ 'abc'* both return a *false* value (the first one should be *true*). If you're using version 1.00 of Turbo Pascal and need to make string comparisons, the following function should help:

```
FUNCTION greater_than(str1,str2 : bigstring) : Boolean;
{
        purpose           returns true if str1 ⟩ str2, else false

        ** NOTE **        bigstring must be a globally defined
                          data type:
                          TYPE
                             bigstring    = string[255];
}
VAR
   len1,len2,indx,count            : integer;
   flag,done                       : Boolean;
BEGIN
   { initialize everything }
   flag := false;
   done := false;
```

```
    indx := 1;
    len1 := length(str1);
    len2 := length(str2);
    count := len1;
    IF count > len2
      THEN count := len2;
    { compare the strings }
    WHILE (indx <= count) AND NOT done DO
        IF str1[indx] = str2[indx]
          THEN indx := indx + 1
        ELSE BEGIN
          done := true;
          flag := (str1[indx] > str2[indx])
        END;
    { this is the check that TURBO misses }
    IF NOT done
        THEN flag := (len1 > len2);
    { return final value }
    greater_than := flag
  END; { of FUNC greater_than }
```

A last comment on Turbo Pascal: The string comparison seems to work fine in the main body of a program. It's just in any procedures and functions that it acts funny. By the time you read this, though, Borland International should have released an updated version. In the meantime, you can use the function above.

String Functions and Procedures. Another useful aspect of the UCSD string standard is that it includes more than just the data type *string*. It also defines a set of functions and procedures that work on strings. Figure 1 lists them, but we'll go over the list one at a time (not necessarily in the order given).

The most commonly used function is probably *length(str)*, which you've already seen in a few of the examples above. It's really just another way of writing *ord(str[0])*; that is, it returns the current length of *str*. This is not to be confused with the maximum length of *str*. For example, the program

```
  PROGRAM length_test;
  VAR
    small_str       : string[15];

  PROGRAM show_length(str : string);
  BEGIN
    writeln('length of ''',str,''' is ',length(str))
  END; { of PROC show_length }
  BEGIN
    small_str := 'hello, there';
    show_length(small_str);
    small_str := 'hi';
    show_length(small_str);
    small_str := ''; { null string }
    show_length(small_str)
  END. { of PROGRAM length_test }
```

produces the output

```
  length of "hello, there" is 12
  length of "hi" is 2
  length of "" is 0
```

concat(s1,s2, . . . ,sn)	returns string composed of all the strings concatenated together
copy(str,index,size)	returns string composed of str[index]. .str[index+size−1]
delete(str,index,size)	deletes ⟨size⟩ characters from str starting at str[index]
insert(str1,str2,index)	inserts str1 into str2 starting at str2[index]
length(str)	returns current length of str
pos(pattern,str)	returns position (index) of pattern within str

Figure 1. String procedures and functions.

The next most commonly used function is probably *concat*. We can use it to patch several strings together. It's handy for inserting string variables in the middle of string constants. For example, this program

```
  PROGRAM concatenation;
  VAR
    name,message     : string;
  BEGIN
    write('Please enter your name: '); readln(name);
    message := concat('Hello, ',name,' how are you?');
    writeln(message)
  END. { of PROGRAM concatenation }
```

produces this output:

```
  Please enter your name: Deirdre
  Hello, Deirdre, how are you?
```

Right up there with *concat* is the function *copy*, which allows us to pull out part of a string. It takes the string, the starting location, and the

length of the substring. The code

```
name := 'Deirdre Ann Wendt';
middle := copy(name,9,3);
writeln('Your middle name is ',middle);
```

would produce

```
Your middle name is Ann
```

Of course, the example above depended upon knowing right where *'Ann'* started in the string. Suppose we knew what string we were looking for but didn't know where it started? We could use the function *pos* to find it. Given a substring or pattern and the string in which to search for it, *pos* returns the location of the start of the substring. If it can't find the pattern, *pos* returns 0. Our code could now look like this:

```
name := 'Deirdre Ann Wendt';
loc := pos('Ann',name);
middle := copy(name,loc,3);
writeln('Your middle name is ',middle);
```

Really, this code doesn't make too much sense, but—what the heck—it's just an example.

The *delete* procedure lets us remove a section of a string. Combined with *pos* and *copy*, it can be used to *parse* a string—that is, to pull off a chunk at a time.

Let's suppose we want to write a procedure that will pull the first word off a string, where a "word" is defined as any substring starting with a nonspace character and followed by a space. Our procedure might look something like this:

```
PROCEDURE parse(VAR line,word : string);
{
      purpose    removes first word in ⟨line⟩ and returns it
                 in ⟨word⟩
}
VAR
  space          : string[1]; { need for some version of pos }
  indx,len       : integer;
BEGIN
  space := ' ';

  WHILE pos(space,line) = 1 DO { remove leading blanks }
     delete(line,1,1);

  len := pos(space,line);     { look for blank }
  IF len = Ø THEN BEGIN{ no blanks left }
    word := line;             { get word }
    line := ''                { zero out line }
  END
  ELSE BEGIN                  { get word and delete from line }
    word := copy(line,1,len-1); { get all but blank }
    delete(line,1,len)          { delete word plus blank }
  END
END; { of PROC parse }
```

Note: UCSD Pascal has an even better routine for finding single-character delimiters. It's called *scan*, and it will look for the first character matching (or failing to match) the delimiter. See the UCSD Pascal documentation for more information.

The last procedure, *insert*, is the reverse of the *copy/delete* operation. It takes one string and stuffs it somewhere inside another. Token expansion is one good use for *insert*. Suppose we were writing a program to take a form letter and put in the appropriate names, dates, and so on. Within the form letter, these fields might be represented by *tokens*; for example, the salutation might look like this:

```
Dear ⟨title⟩ ⟨last name⟩:
```

The program would scan through the form letter, list all the tokens (which are all marked with the ⟨ ⟩ characters), then get the information to replace them (interactively, from a file, or whatever). The following procedure, then, might be of use:

```
PROCEDURE replace(VAR line : string; token,sub : string);
{
      purpose   look for token in line and replace with sub
}
VAR
  indx,len        : integer;
BEGIN
  REPEAT
    indx := pos(token,line);
    IF indx > 0 THEN BEGIN
      delete(line,indx,length(token));
      insert(sub,line,indx)
    END
  UNTIL indx = 0
END; { of PROCEDURE replace }
```

The following code

```
line := 'And so, ⟨title⟩ ⟨last⟩, the entire ⟨last⟩ family';
replace(line,'⟨title⟩','Dr.');
replace(line,'⟨last⟩','Lewis');
writeln(line);
```

would produce

```
And so, Dr. Lewis, the entire Lewis family
```

This should give you a clue about how all that "personalized" junk mail that you receive is generated. And you thought that all those people cared just about you!

Conclusion. You should have a lot to play with for a while. Good. We'll be absent next month, owing to preparations for the West Coast Computer Faire. When we come back in June, we should have our long-delayed review of Modula-2 (our apologies to Volition Systems!). See you then. ▲

THE BASIC SOLUTION

by Joe Juhasz

Peeks and Pokes: Keyboard Status and Other Bits

This month we're going to continue the survey of interesting peeks and pokes that we began in our last installment (February 1984). In particular we'll look at the memory byte that holds shift-key status information, and we'll present a method for clearing the keyboard buffer by way of a *poke* statement.

Information concerning the status of the PC's four shift keys and four toggles is stored at address 0000:0417—or, in other words, at an offset of 417H in segment 0. The four shift keys are the right and left secretarial shift keys (the ones with the fat arrows), the control key, and the alternate key; the toggles are scroll lock, num lock, caps lock, and insert. Since each of these keys can only be in one of two states, the PC is able to record the status of each key with a single bit.

Here are the bit assignments of the keyboard status byte:

Bit 7:	Insert
Bit 6:	Caps lock
Bit 5:	Num lock
Bit 4:	Scroll lock
Bit 3:	Alt shift
Bit 2:	Control shift
Bit 1:	Left shift
Bit 0:	Right shift

For the shift-key bits, a 1 means the key is pressed; for the toggles, a 0 represents the toggle's default condition.

Naturally, the status bits function independently of one another; so if, for example, you hold down the left and right shift keys while in insert mode (with the other three toggles in their default states), a peek at &H417 (following a *def seg = 0*) will return the value 131 (1000 0011B).

Isolating Keyboard Status Bits. In order to read or set an individual keyboard toggle (or shift key), without affecting any of the others, you need to isolate individual bits within the keyboard status byte. To do that, you can use the logical operators, *and*, *or*, and *xor*.

For example, to turn the insert toggle on, you can *or* the keyboard status byte with 80H (1000 0000B). Regardless of what's in location 417H to start with, the result of this *or* operation will have a 1 in bit position 7:

	0110 0000
OR	1000 0000
EQUALS	1110 0000

The following BASIC code, then, will set the insert toggle:

```
DEF SEG = 0
KEYBOARD.STATUS = PEEK(&H417)
KEYBOARD.STATUS = (KEYBOARD.STATUS OR &H80)
POKE &H417,KEYBOARD.STATUS
DEF SEG
```

To turn one of the other toggles on, just change the value of the *mask* (the second operand of the *or* operation) so it has a 1 in the appropriate bit position.

Let's say you want to turn caps lock off. For this you can use the *and* operator. The caps lock toggle status is reported by bit 6 of the status byte, and you want the result to have a 0 in that position. So your mask *this time will* have a 0 at bit 6; if you want to leave the other status bits alone, put 1s in all the other mask bit positions. Here's an example of the logical operation:

```
           1110 0000
AND        1011 1111
EQUALS     1010 0000
```

And here's some Basic to do it:

```
KEYBOARD.STATUS = (KEYBOARD.STATUS AND &HBF)
DEF SEG = 0
POKE &H417,KEYBOARD.STATUS
DEF SEG
```

Finally, to toggle a status bit—that is, to turn it on if it's off and off if it's on, use *xor*. For this example, let's say you want to change the status of num lock (bit 5). The logic looks like this:

```
           1110 0000
XOR        0010 0000
EQUALS     1100 0000
```

And your Basic code might look like this:

```
KEYBOARD STATUS = (KEYBOARD.STATUS XOR &H20)
DEF SEG = 0
POKE &H417,KEYBOARD.STATUS
DEF SEG
```

Purging the Keystroke Buffer. Addresses 0000:041A and 0000:041B contain a pointer to the head of Basic's keystroke buffer. The lower of these two addresses (0000:041A) contains the offset component of the buffer head address, and the upper contains the segment component. Similarly, 0000:041C and 0000:041D point to the tail of the keystroke buffer. By setting the two pointers so they point to the same place, you flush the keystroke buffer. Since the segment components of the head and tail addresses are already equal, you only have to concern yourself with the offsets. Here's the code:

```
DEF SEG = 0
POKE &H41C,PEEK(&H41A)
DEF SEG
```

Another, less efficient way to clear the buffer is:

```
1000 IF INKEY$ <> "" GOTO 1000
```

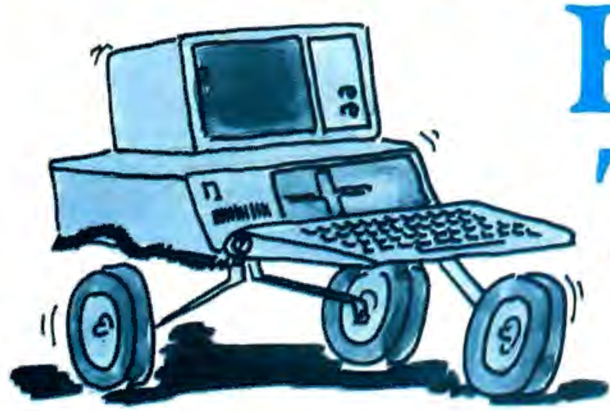
The latter method clears the buffer one keystroke at a time; the former does it all at once.

Reading the BIOS Release Date. One more tidbit this month: The release date of your ROM BIOS is stored at F000:FFF5. The following Basic statements retrieve and display this date:

```
DEF SEG = &HF000
FOR I = &HFFF5 TO &HFFFC
PRINT CHR$(PEEK(I));
NEXT
```

More peeks and pokes in issues to come. We do take requests; send them to Softalk/IBM, Basic Solution, Box 7040, North Hollywood, CA 91605. ▲

Put Training Wheels On Your IBM PC™ Before You Take Your First Ride.



What is the easiest, most affordable way to learn how to use your new PC?

Well, you could read tedious DOS and operating manuals or you could enroll in a two day course for a few hundred dollars. But why go through the frustration of reading or the expense of a course when there is a better way?

The easiest, most affordable way to learn

The answer: **NO. 1 HOME TUTOR**

No. 1 Home Tutor combines completely self-paced instruction, interactive lessons and comprehensive training to get you using your PC quickly, easily.

You Learn at Your Own Speed

No. 1 Home Tutor is your learning companion. No pressure. No frustration. The lessons are completely self-paced — you learn as much as you want, when you want to. Each lesson can be stopped at anytime and restarted without having to start over from the beginning. No other training program allows you to do this.

The Lessons Work with You

No. 1 Home Tutor is a hands-on tutorial. You participate in the simple lessons to learn more and retain what you learn better in less time. You interact with each lesson, learning and practicing the operations of your PC.

No. 1 Home Tutor is a registered trademark of MA Systems, Inc.
IBM PC is a registered trademark of International Business Machines, Inc.

The Easiest Training Program Available

With No. 1 Home Tutor, you don't have to read manuals — simply follow the instructions on the screen. The lessons are visual — providing multicolor displays that show what you are learning. You practice and learn by interacting with the screen. It's that easy.

Comprehensive Learning

No. 1 Home Tutor teaches you the simplest operations to the more advanced. No. 1 Home Tutor shows you how your computer and programs operate and how to use them. After using No. 1 Home Tutor, you will be amazed at how much you know about using your computer. You'll be ready to use your computer right away.

Best Value

For only $59.95, no other training program available makes learning easier, more comprehensive and affordable than

NO. 1 HOME TUTOR.

That's why it's called "No. 1."

So learn about your PC the easy way, with **NO. 1 HOME TUTOR.** Ask your local computer dealer for a demonstration.

No. 1 Home Tutor is made by MA Systems Inc., respected manufacturers of quality hardware and software products for IBM and Apple computers.

MA SYSTEMS INCORPORATED An Enconi Company

2015 O'Toole Avenue, San Jose, CA 95131 (408) 943-0596

Late Sports

In The News

Today's Weather

Volume II Issue XI April 1984 Softalk

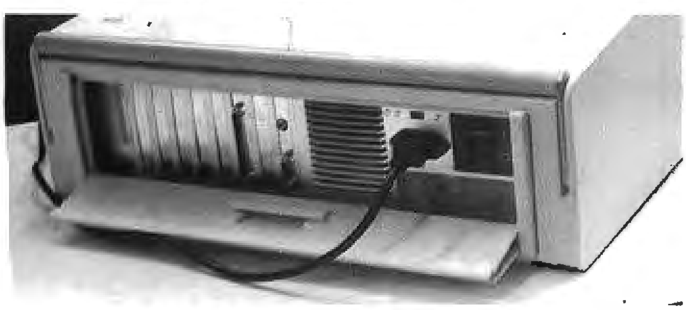
IBM Announces Portable PC

On February 16, IBM announced its latest addition to the PC family: the Portable Personal Computer. The long-awaited PPC is a transportable (thirty-pound) machine measuring 20 by 17 by 8 inches. It sports a built-in nine-inch amber display, one or two horizontally oriented half-height ~~double-sided~~ floppy disk drives, 256K (that amount is standard), and five expansion slots (three full-length slots and two shorties). The system board is described by IBM as identical to that of the XT.

The keyboard, identical in layout to the PC's, attaches to the front of the PC by way of a telephone-type connector. In travel mode, the keyboard snaps in over the display and disk drives and forms the base of the machine as you carry it.

On the posterior side, a handle is attached to a sort of folding door. When the PPC docks at your desk, this handle and door move out of the way to reveal the power switch, the exhaust port for the internal fan, and the connectors for whatever I/O cards you have installed. The power supply accommodates either 220 volts or 110.

The display is driven by IBM color/graphics adapter circuitry, which is built into the PPC's system board (in other words, the display does not consume one of the system's five slots). Unfortunately, there's no way to make the PPC put text on an IBM monochrome display. You can't simply plug in a monochrome adapter card and attach a monochrome display;



the latter normally attaches to a power socket on the PC or XT system unit, and that socket doesn't exist on the PPC.

The IBM Product Center price for the Portable PC equipped with two disk drives and 256K is $3,220; add $150 to that price if you want the IBM parallel printer adapter card.

The Portable PC was due to arrive at retailers around mid-March; in the Los Angeles area, the IBM Product Centers had their demonstrator models by that time but other retailers did not. By mid-April, you can expect the machine to be available in very limited quantities at most stores that sell XT, PC, and Junior.

At the time of its debut, this machine was widely described (by spokespersons for Compaq, among others) as a Compaq look-alike. Indeed, the size and general appearance of the two machines are quite similar. But there are significant differences (for more about the Compaq, see "Close Up with the Compaq," September 1983).

The Portable PC's inability to display the monochrome adapter character set—either on its built-in amber display or on an externally attached monitor—is the major difference between the two machines; it's also a clear liability for IBM. If IBM had to choose one display adapter or the other, then of course they chose correctly; without graphics capability the machine would have drawn more hoots than PC*jr*'s keyboard. But Compaq managed to put the functionality of both display adapters into a single board driving a single CRT.

The Portable PC's keyboard provides the same tactile and aural feedback as does the PC's. Whether you regard that as a plus or a minus relative to Compaq's Key Tronic board is likely to depend on your typing style. The IBM keyboard pushes back harder against your fingers and therefore may be more suitable for a percussive typist. In any case, the fact that the IBM keyboard attaches by way of a loose cable to (and detaches from) the area between the display and the disk drives is an unqualified advantage for IBM. The Compaq keyboard cable is a stiff coil, and, because it emerges from the left side of the front panel instead of from the middle, it tends to pull the corner of the keyboard away from you; a more serious drawback to the Compaq design is the fact that the keyboard can't be removed.

In standard minimal configuration—that is, with 256K, two disk drives, and a parallel printer port—the Portable PC appears to have one more available slot than the Compaq.

The IBM machine also sells for about $200 less than a comparably equipped Compaq. Clearly Big Blue is trying to put some price pressure on Compaq and the other makers of portable PC work-alikes. Whether that will result in any price reductions from the latter camp remains to be seen. It also remains to be seen, of course, to what extent IBM will be able to deliver its new machine. Odds are good that sales of compatible portables will not be seriously affected until IBM is able to get a lot of PPCs into the stores, and that could take half a year or more.

Lotus 'Symphony' Premieres

Mitch Kapor Introduces Followup Act to 1-2-3

Mitch Kapor has been hearing a lot of good music lately, and on a sunny Valentine's Day in New York City he played some of it back to his public. Kapor opened the concert with a sweet song of success, announcing Lotus Development Corporation's $53 million of revenues for 1983 (the company's first full year of operation). Those revenues translated to $13.7 million in net earnings—$.98 per share of Lotus's recent public stock offering.

Symphony followed song, as Kapor announced Lotus's much-anticipated new product. *Symphony* is an integrated spreadsheet-graphics-word processing-database management-telecommunications package that Lotus calls "the first complete business productivity tool ever offered for the desktop microcomputer market." Kapor says that the combined facilities of the program make it "so wide-ranging, comprehensive, and easy to use that computer users can turn it on in the morning and leave it running all day."

Symphony is indeed wide-ranging. It combines all the features of the astoundingly successful *1-2-3* (the basis for the company's fortunes to date) with a word processor touting "Wang-like features," a database management system that includes a mail-merge feature and a forms-oriented user interface, and asynchronous communications.

Symphony's polyphonic weave all takes place within its own "window" environment, and the work also features help screens, an on-disk tutorial, extensive documentation, and a programming language (*Symphony Command Language*) for turning repetitive tasks into procedures. Lotus's highly rated customer-support service will be available to provide further assistance when the product is released—"in midyear."

Combining word processing with spreadsheet and database capability is not new in the microcomputer industry, but it's still an emerging technology. And Lotus seems to have done a good job of it. All *Symphony* functions and data remain resident in RAM, so the product is very fast; *Symphony* switches functions and data almost instantly. The design, however, is extravagant in its use of memory; a minimum of 320K is required, and 640K is recommended for sophisticated applications.

It is through the window environment (which is controlled by cursor key, not by mouse) that the user accesses *Symphony*'s functions and sees results. This environment allows simultaneous or sequential viewing of any of the functions or data being worked on; among the capabilities provided are a "zoom" key for changing the size of any window and multiple "tiling" of windows. Because all data is resident in memory, Lotus's windows operate exceedingly fast.

Lotus has improved on the already good *1-2-3* spreadsheet by making it larger (8,192 rows by 256 columns—the largest spreadsheet in the industry), enhancing character-data manipulation, and adding a degree of data security. Data can be stored in the spreadsheet format or in the *Symphony* database. The interfaces for *dBase II* and other database products that were developed for *1-2-3* have been brought into the new product.

Symphony's own database is "forms-oriented," which means it will be easy for database novices to use. It can handle applications requiring up to about eight thousand records. The integration of spreadsheet and database gives the user access to an easy and fast database report generator (the spreadsheet), and the integration of database and word processor makes the generating of management reports, which usually requires lengthy explication of all those numbers, a snap.

Symphony's word processor is alleged to have "Wang-like" capabilities and intuitive commands; best of all, it can integrate the calculations and graphs made in a *Symphony* spreadsheet, or data from the database, into a document. A user can update a document containing merged data while simultaneously updating the spreadsheet from which the data was merged—and view both activities at once in separate windows. The mail-merge feature (provided in conjunction with the database) can also be used to control the merging of separate spreadsheets.

Kapor describes the product as "an open-architecture design," which means that modular additions should be feasible; in fact, software vendors are being invited to bring additions to Lotus for consideration. Eventually, users will be able to add their own modules, meaning that *Symphony* could be a dark-horse candidate for the IBM PC operating system of the future.

Symphony is not *1-2-3* cum revisions; it was written from the ground up, which accounts for its outstanding performance. But it's fully upward-compatible with *1-2-3*.

That's not all the good news for present Lotus users, however. Registered customers can trade in their *1-2-3*s for copies of *Symphony* by paying only the difference in price. That's a unique policy in the software industry, and, given *Symphony*'s $695 price tag, it's a particularly happy circumstance for present customers. The trade-in arrangement is apparently Lotus's way of paying back those who made the company so successful. It's a good bet that at least 75 percent of current Lotus customers will take advantage of this offer.

Neither price nor memory requirement is expected to be a barrier to *Symphony*'s sales potential. The product is clearly aimed at the large corporate marketplace (Lotus has even established its own corporate sales staff, and high-volume *1-2-3* customers were at the company's offices in Cambridge, Massachusetts, viewing the new product during the week before

Mitch Kapor of Lotus

the announcement.) Most corporate PCs have at least 256K installed, and having to add a memory expansion board is not a stiff penalty in light of *Symphony*'s productivity potential for middle managers.

The first version of *Symphony* will be available no sooner than midyear ("That's July second, according to *Symphony*'s date routine," quipped Kapor) and will be offered only for the IBM PC. Versions for other computers will be along in the future. Kapor stated that the 3270-PC would **definitely** be supported in an upcoming release of the product. ▲

Multi-User dBase II Unveiled

dBase II has always defined relationships among data, and now it can monitor the relationships among multiple users of that data. Ashton-Tate has introduced a multiuser version of its bestselling database management system that allows an unlimited number of users to share a *dBase II* database and application through a local-area network.

Describing *Multi-User dBase II*'s lockout feature, which prevents users from colliding, as a "traffic cop," Harvey Jeane, vice president of product development for the Culver City, California-based firm, called the product "ideal for the office that needs centralized data storage with the flexibility of individual workstations." The lockout feature prevents one user from writing to a file or record that's being written to by another user or users.

The new version of *dBase* was introduced February 15 at New York's Tavern on the Green, using 3Com's Ethernet local-area network system. Ashton-Tate said they initially selected 3Com because of that company's willingness to work with them in developing the new product; they will, however, make the system available on other networks as those networks become available and widely used.

The software (including *dBase* and the application program) resides on the network's Winchester disk and becomes resident in a workstation's memory only when called for. The data always remains on the central network file.

Multi-User dBase II provides data lockout at the file, index, and record levels. The lock is only for write permission. Any user validated for the network can read any record or file at any time, making the system far less sophisticated than its time-sharing-based cousins.

Licensing for the new product is interesting and may provide a hint of things to come from other players in the network arena. *Multi-User dBase II* can be purchased in units accommodating four users. Each unit allows four users to be logged on simultaneously, but any number of users may be validated for the application or system.

The new *dBase* incorporates a protection scheme that prevents the program from being copied from the network disk to a user's local disk or diskette. Ashton-Tate is extending this protection scheme to its normal (single-user) *dBase II* as well and may extend it to *dBase II* applications if necessary. The protection mechanism allows for full backups of *dBase II*, databases, and applications.

Pricing for the new product is straightforward. A starter kit for four users can be purchased for $1,000, and additional four-user units cost $500 apiece. Current *dBase II* users can purchase an upgrade to the multiuser version for $300. The protection scheme used by Ashton-Tate does the validation of new users as well, upgrading only the copy resident on the network when installed.

No other new features have been added to *dBase II*. When asked about availability of *Multi-User dBase II* for machines other than the IBM PC, Ashton-Tate said only that things were under consideration. ▲

No other financial software even comes close.

Now Financier II
The financial software that has met with resounding approval from professional software reviewers just got even better. Now, the Financier Personal Series has been upgraded to Financier II. Combined with Financier's Tax Series and Investor's Series, it creates a totally integrated financial management system. Financier II is so easy to use that it is ideal for personal finance yet powerful enough, and flexible enough, to function as a small business system.

"It is clear that the Financier Personal Series is the most sophisticated and flexible of the financial programs under review here."*

Burton Alperson, Ph.D.
Reviewer, SOFTALK
June, 1983

Financier II features enhanced graphics capabilities, and expansion of Financier's unique auto-checking capability which prints checks directly from a user-maintained database, while updating bank, budget and tax records. Additionally, it provides a full range of professional-caliber financial statements including net worth, and cash flow reports. You can even opt for double-entry accounting methods, if desired.

Introducing Financier II

Now you can have a completely integrated financial software system.

"Out of all these programs, the Financier Personal Series provides the best balance of extensive capabilities and ease of use. The manual is outstanding as an introduction to both the practice of bookkeeping, and the use of a personal computer to perform this task."**

Steven Yoder and
Sherry Knight, CPA
PC Magazine, February, 1983

Financier II is remarkably easy to use.
You don't have to be a computer whiz-kid to use Financier II. That's because it's menu-driven. When you need to enter data, just fill in the blanks on the pre-formatted screen. If you can use an automatic teller machine, you're well on your way to mastering this unique financial software.

Another thing that puts Financier II so far in front of any other financial software package is that it offers powerful tax and inventory functions. In fact, Financier II allows you to maintain literally thousands of these accounts, if you desire. And, each account is separately manageable.

"Thus far, no other software company has matched Financier's interlocking combination of a powerful tax-planning package (Tax Series) and a year-round record-keeping program (Personal Series)."

Augustin Hedberg
Money, June, 1983

Financier Tax and Investor's *Packages.*
Financier's Tax and Investor's Series provide the perfect adjunct to Financier II.

Financier's Tax Series lets you create multiple tax cases, and store them for later adjustment. You can even perform "what-if" spreadsheet calculations, generate graphics and print charts.

Financier's Investor Series is a comprehensive portfolio management system that lets you maintain a data base on each security account, each security position, and each security transaction.

Financier Offers Something for Everyone.
Individuals, businessmen, tax preparation specialists, investors or anyone who wants real financial control cannot afford to miss out on Financier's powerful software packages.

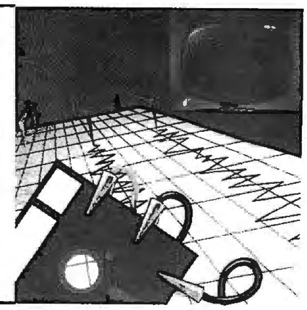
Get the most from your personal computer with the financial software that does the most. Financier II, and Financier's Tax and Investor's Series.

FINANCIER
WE'LL KEEP YOU ONE STEP AHEAD ®

Financier, Inc., 2000 West Park Drive, Box 670, Westboro, MA 01581 (617) 366-0950

*Apparant Home Finance™, Perfin™, PC/PFP™, Home Accountant Plus™, Financier Personal Series™
**Bonus Accounting System™, Financier Personal Series™, FMS II™, Home Accountant Plus™, Money Maestro™
Financier software is available for IBM® PC, IBM® XT, Digital Rainbow™ and Wang® PC

marketalk reviews



Unless otherwise indicated, software listed runs in DOS 1.1 or 2.0 with either display adapter and requires 64K and at least one disk drive.

Pack & Crypt

Pack is a file-compression program that can save storage space for archival data. Text and source-code files so treated take up approximately 30 percent less disk space, with somewhat greater compression for data files. Since application programs can't interpret the compressed data, a companion routine expands the files back to their original size and format. For users who pay list price for disks, have time on their hands, and rarely look at old files, *Pack* might save a buck here and there in file storage.

Crypt is a file encryption/description program. Although the program is not certified under the NBS encryption standard, its authors feel it is secure for most purposes. *Crypt* uses a fourteen-character keyword as a basis for encryption. The same keyword is required to decipher the files.

Included in the *Pack & Crypt* set is *Hexer*, a utility that translates files that have been converted by *Pack* or *Crypt* into pairs of hexadecimal digits. These can be sent as seven-bit codes by modem; then *Hexer* must be used at the other end to translate the files back to their previous form. Since this program will send two bytes for every one in the original file, your phone bill is bound to go up. Users might be better off sending eight-bit words over their modems rather than using *Hexer*.

Clearly, *Crypt* is the best of the trio. If you store files where the wrong people can snoop and play with them, *Crypt* could be the better mousetrap you've been looking for.

Pack & Crypt, Norell Data Systems (3400 Wilshire Boulevard, Los Angeles, CA 90010; 213-257-2026). $49.95.

—*Dan Robinson*

Realfast 1

For those of you who have been struggling with the IBM's Fortran compiler (by Microsoft), an interface to the 8087 numeric coprocessor is now available. For PC owners who have installed the 8087 chip, GSII, the maker of *Realfast 1*, offers to increase the speed of mathematical calculations by as much as 600 percent and of formatted input/output by up to 500 percent. While the package may fall short of these claims when used with many Fortran programs, there is no reason to doubt that such increases could be obtained in selected programs.

Realfast 1 is a module combined with the other compiled Fortran modules by the standard IBM Link program. This module replaces the library subroutines that normally handle real (floating-point) numbers for arithmetic operations, comparisons, trigonometric functions, exponents, and logarithms, as well as conversion of these numbers to ASCII characters for formatted I/O. The routines being replaced are slow and require a lot of memory. By allowing the numeric operations to be handled by the 8087 chip, this package reduces the total memory required for a compiled Fortran program by approximately 6K. In addition, the 8087 does most of the previously mentioned processes 80 to 200 times faster than the library routines. That's the good news.

The bad news is that most programs spend a lot of time moving numbers around the registers and memory and doing other work that isn't related to the 8087. So even if your old program spent 90 percent of its time doing numeric calculations, it could be made only ten times faster even if the 8087 were infinitely fast. The other bad news is that IBM Fortran does not store its real numbers in a format the 8087 can use. In other words, the way a real number is represented in 0s and 1s by IBM Fortran is not the same way the 8087 expects to see them. IBM Fortran uses a Microsoft representation, while the 8087 uses the IEEE floating-point standard. This incompatibility is overcome by *Realfast 1* but at a sacrifice of speed. When run on a Whetstone benchmark program (see "Good News on the Fortran Front" in the November 1983 *Softalk*), the *Realfast 1* package improved execution speed by six times (503 percent). However, the same program compiled with Microsoft's Fortran 3.1 compiler (which uses the IEEE format) was eleven times faster (1,050 percent).

When a comparison was run with a GSII-supplied benchmark program, the Microsoft Fortran was slightly better in reading and writing 200 formatted fifty-byte records and in performing 1,000 trigonometric/logarithmic operations. However, in performing 10,000 arithmetic operations, the Microsoft Fortran was more than two and half times faster than the IBM Fortran with *Realfast 1*. For those looking to buy a Fortran compiler, this package is not a good choice. Microsoft's Fortran compiler offers high-speed execution along with a much better interactive I/O capability. It's also less expensive than IBM Fortran without *Realfast 1*. However, if you already have the IBM compiler, the improvements offered by this package are significant. Although the documentation supplied is very limited, the program is easy to use. One word of advice: Be sure to use an updated (patched) copy of the IBM Fortran compiler.

GSII offers similar packages for IBM's Pascal compiler, Basic compiler, and Basic interpreter. Since the real-number routines of these languages seem to be faster than those of IBM Fortran, the speed improvement afforded by the GSII packages may not be as great.

Realfast 1, GSII (Box 1193, Golden, CO 80402; 303-277-0073). $120.

—*Jack Wilschke*

Persuasion

Persuasion is a monochrome maze game for one or two players. As with most other games of this type, the ballyhoo on the package features lavish descriptions that are far removed from the simple displays. The game itself consists of different mazes, which are surrounded by borders, or "walls." One or more Greek characters are displayed along each wall, and the object of the game is to traverse the maze, avoiding the hordes of reverse-video numbers that represent hostile factory workers, and reach a blinking character that represents a "Key Weapons Module" within a weapons plant. Plant executives, call "War-Mongers," roam the maze in the guise of dollar signs and try to impede your game piece.

The game does have some interesting features. Hostile workers are represented by numbers 1 through 9 and friendly workers by letters A through I. (The higher the number or letter, the more hostile or friendly the worker.) Hostile workers pursue the game piece and attempt to sur-

round it, but you can change the lower numbers into "friendly" letters by ramming them repeatedly.

There are three main types of mazes: horizontal/vertical, honeycomb, and diagonal. The first two require that the player go through the maze corridors or find doorways. The maze built of diagonal lines allows the player to pass between the diagonal blocks. Another type of maze combines horizontal, vertical, and diagonal walls, which results in some interesting movement restrictions.

Since strategy for this game depends a good deal on selecting the best path through the maze, the designers have obligingly provided a "freeze" function. When you press the escape key, all action stops and you can leisurely analyze the position.

You can build walls to stave off the enemy, and you have occasional opportunities to reach a "portal" character, which adds more friendly workers to the scene while displacing enemies. A "teleport" capability moves your piece to another section of the maze at the cost of points.

Scoring is based on "resolution" and "insight," which continually change during play. The more hostile workers surround your piece, the weaker your resolution becomes, as shown by a vertical bar on the right of the display. Resolution is always restored to full strength when one of the goal characters is reached. Insight depends on movement toward the goals and is carried over from maze to maze.

Persuasion can be played with joysticks or keyboard. If two players want to use the keyboard, the second player is forced to use a pseudo-pad formed by the QWEASDZXC keys. While this lets both players sit at the keyboard, it's a rather awkward device.

Persuasion is not a graphics game. It uses the regular PC character set for game pieces. The company says the game does operate on a color display, but since it is self-booting there is no way to switch from mono to color if you have both displays. Play is rather slow, even at the highest speed setting, and the price seems high for a simple maze game—notwithstanding its socioeconomic bias. The concept is good, but somehow this game lacks that essential pizzazz.

Persuasion, by Stephen Linhart, Olorin Games (Box 719, Amherst, MA 01004; 549-413-0535). $40.

—Dian Crayne

Graphics Utility

Graphics Utility is one of those packages that falls through the cracks. It's a little simplistic for an experienced programmer and a bit confusing for a novice. The four graphics programs on the disk let you create small medium-resolution figures, save them, combine them, and use them in simple animation. The programs are written in interpretive BASIC and aren't copy protected, so you can make backup disks easily and can study the code to see how it operates.

Graphics Utility uses a grid approach for its drawing, so you fill in the squares with any of the three available palette colors. The *Graphics Utility* grid can be from 1 by 1 to 39 by 39 medium-resolution pixels, which allows for good control in the design of characters.

The character-generation program, Chardraw, prompts for the grid size and the name of the file where you want the figure to be stored. The program then assigns a number to the figure, depending on the order of generation; this number is used by the other programs to locate the figure in the file. Figures can be created or edited, but there is no provision for deletion. They also can be copied and stored under new names, which makes creating animated sequences easy.

The cursor can be moved only horizontally and vertically when you are drawing a figure, since the 7, 9, 1, and 3 keys on the ten-key pad are used to move the cursor directly to the four corners of the grid. Each square passed over by the cursor is filled in automatically, and you can draw figures rapidly. A reverse function swaps the filled-in squares to the background color or back again, but you have to use this function with care when you're working with multiple colors. An actual-size display of the figure appears at the top right of the screen as you draw on the grid, allowing you to see how the character will look on the color display.

Charanim is the animation program. It accepts up to eight-figure numbers, prompts for movement speed, and then displays the figures in sequential positions across the screen, left to right, creating animated movement. There is no way to create diagonal movement or change the movement direction, but the speed can be varied, and Charanim also prompts for the palette.

The Chargrup program combines several of the small characters into one large figure. Using it requires a little planning, since you have to enter the horizontal and vertical position for each character to be used. The resulting figure can be stored and later retrieved with the Chardisp program, which displays only grouped figures. If a single character is to be displayed, it must be grouped singly; there's no other way to show it off the grid.

Four subroutines included in the package can be incorporated into your own programs; these provide for storing a character in an array, reading it back out, and displaying it on the monitor screen, as well as for displaying characters in sequence as an animated routine. These subroutines give you the flexibility of creating figures and then reading them into an array to be used in games or presentations.

Each subroutine has a comment section that tells you what values to insert in the code. The code itself is not commented, however, and would be hard for a beginner to understand. The longest routine is ten lines of code long, and the shortest is only two. All the routines are simple enough that an experienced programmer could write them from scratch and have all four up and running in an hour or so, but beginners will find them useful.

The manual is small and obviously intended for users with experience in BASIC graphics. The information is straightforward and com-

plete but not sufficient for inexperienced users. A demo file on the disk is referred to as part of a tutorial, but the manual switches confusingly back and forth between discussion of the demo files and advice on creating new figures.

Graphics Utility has limited capabilities, but it does what it does quite well. The program is attractively priced, and it could be handy for people who have the need or desire to draw many small, medium-resolution characters.

Graphics Utility, by Savant Software (Box 440278, Houston, TX 77244; 713-556-8363). $85.

—Dian Crayne

Ultima II

The *Ultima* games by Lord British have long been among the most popular adventures on the Apple. Now, finally, one has been translated for the IBM.

This isn't an adventure like *Zork*, with lots of text, no graphics, and plenty of time to make decisions. It's more like a computerized *Dungeons & Dragons*, where you kill monsters and plunder treasure ornamented by sharp graphics, whiz-bang animation, and sound effects.

As in *D&D*, you start the game by creating a character that reflects your game strategy. For example, fighters depend on muscle and cutlery to solve problems. Thieves sneak around misappropriating things. Wizards and clerics rely on magic. Your character will be endowed with six attributes—strength, agility, stamina, charisma, wisdom, and intelligence, and you have ninety points to distribute between them. In addition, you get a bonus for your profession (fighters get a strength bonus; thieves get an agility bonus). There are bonuses for sex (men are strong, women more charismatic). And you get a bonus for the race you choose; humans, elves, dwarves, and hobbits each have their own advantages.

Your character materializes in North America—not the continent you're familiar with, but the North America of 1423 B.C. There are, at least, no Indians and grizzly bears to worry about. Instead, you must confront the myrmidons of the evil sorceress Minax. After fighting her orcs, demons, and pirates for a while, you'll wish you had only a simple grizzly bear to deal with.

Your character appears as a high-res figure on a colorful map of the continent. (The display is not nearly as attractive in black and white, and if you don't have a color adapter, forget it.) Your hero is always in the center of the screen, so if you move toward Alaska, say, the map scrolls around you.

Dotting the map are a number of features—towers, dungeons, villages, towns, and castles. The towers and dungeons you can explore in search of adventure. (Actually, it is possible to win the game without going near them.) The villages and towns are indispensable. There you buy equipment, acquire spells, and pick up clues. In the castles, you offer tribute to the world's rulers and receive their bounty.

Occasionally, a time door will pop up (as in *Time Bandits*). Enter a time door, and you could reappear almost anywhere, from ancient Pangea to post-W.W.III America. Most sinister of all is the time of Legends, full of powerful demons and unforeseeable perils. But if you pop through time doors blindly, you'll soon come to grief. Fortunately, the game includes an attractive map with useful clues for the traveler.

As the game progresses, your character can improve in almost every way. Your attributes can increase. You can purchase—or steal—weapons and armor. You'll accumulate a variety of miscellaneous possessions—keys, tools, magic wands, mysterious gems. You will learn to use all sorts of transportation, from foot to horse, from frigate to airplane to rocket ship. You eventually will probe the nine planets of the solar system, all the way to Pluto. Most of all, you'll gather the clues, tips, and oddments of information you need to understand how the game works. Finally, you'll confront the greatest source of evil, and take a stab at changing history itself.

All hail, Lord British! *Ultima II* is the most sophisticated adventure program yet available for the IBM.

Ultima II, by Lord British, Sierra On-Line (Sierra On-Line Building, Coarsegold, CA 93614; 209-683-6858). Requires color/graphics adapter. $59.95. Separate version available for PC*jr*, $59.95.

—Forrest Johnson

J-Bird

Arcade denizens will remember *Q *Bert*, an engaging, big-nosed fellow who spent his entire existence hopping around on a pyramid, changing his color, while dodging bouncing balls and a predatory snake. With the release of *J-Bird*, old *Q *Bert* has grown a tail and migrated over to the IBM, closely pursued by snake, balls, et al.

Greg Kuperberg gets the credit for this competent, though not too original, game. The machine language graphics are sharp and reasonably fast. However, since the keyboard buffer is disabled, you do have to hit the same key twice on occasion—a minor annoyance that can be remedied by the use of a joystick. On a black-and-white monitor, also, it's sometimes impossible to distinguish the "colors" you are supposed to be changing. Other than that, you might almost swear you were playing the original arcade game.

As in *Q *Bert*, your hero starts at the top of the pyramid and works its way down. Every block the bird touches changes color. Once it has altered every block, it moves on to the next pyramid. But things aren't quite that easy for our little birdbrain. The aforementioned snake stalks it, and the balls give it a hard time. On later pyramids, J-Bird encounters new dangers—frisky black cats and a mean frog that changes the colors back again. However, the frog is edible, and "wings" on the edges of the pyramid will give the bird a quick trip to the top if it jumps

on them (perhaps causing the pursuing snake to take a fall). All of this is punctuated by sound effects—for example, a falling sound when something goes off the edge of the pyramid, and an angry chirping when something glomps the bird. If the bird manages to survive hopping around the three pyramids, it proceeds to the next level, where it must convert every block twice. (And after that three times!)

J-Bird has the friendly features we've come to expect from any professional arcade-type game. There's a pause key, so you can answer the phone. The sound can be turned off, so you can talk on the phone and play at the same time. High scores are saved to disk. There are four levels of difficulty. On top of that, there is a competition mode for two players (but only one bird on the screen at a time).

J-Bird, by Greg Kuperberg, Orion Software (Box 2488, Auburn, AL 36831; 800-821-8088). Requires color/graphics adapter. $36.95.

—Forrest Johnson

Media Magician

Media Magician is a specialized full-screen editor that allows you to view or directly edit the contents of any disk sector in either hex or ASCII codes. Disk blocks may be addressed by their physical location on the disk or by their relative location in a file. A number of other capabilities are also offered, such as examination of hidden files and file allocation tables. *Media Magician* takes full advantage of the memory-mapped video display of the IBM PC, and screen updates are extremely fast.

The program is easy to use and has good on-line help menus, so your reference to the manual may be completely unnecessary. This program could be quite useful for *small* changes to a file for an experienced programmer who is comfortable with both the binary machine codes of the 8088 and ASCII representations of text. If extensive patching to a program is required, you would be better off using the interactive assembler that is built into Debug.

Three small criticisms: First, the program is much too large for what it does (it's almost 33K, which may not seem like much until you realize that that's larger than PC-DOS itself). Second, the set of commands available in edit mode is unnecessarily weak. The only cursor commands available are up, down, left, and right. The program would be considerably more powerful if the author had also implemented the home, end, page-up, page-down, enter, backspace, and tab keys. And third, *Media Magician*'s screen format works improperly (for unclear reasons) when asked to display blocks numbered higher than 26330 on a hard disk.

Media Magician, Photon Software (636 120th Avenue N.E., Bellevue, WA, 98005; 206-451-8272). $48.50.

—Ray Duncan

Backgammon 5.0

If your PC's complaining about a constant diet of *dBase II* and *VisiCalc*, maybe you should introduce it to *Backgammon 5.0*, a game that requires both the luck of the dice and a high degree of skill. But don't expect your IBM to be an easy opponent; at its highest playing level, *Backgammon 5.0* can put most humans to shame.

This is one game you can play even if your PC is the all-business monochrome variety. It's more impressive, though, in technicolor. After you set your computer's skill level—on a scale of 1 to 9—the game begins with an on-screen roll of the dice. If the computer wins, it automatically makes the first move. When it's your turn, you hit return to roll the dice; then enter the point number (they're numbered 1 to 24) of the man you wish to move. After making sure your move is legal, the program automatically moves your man according to the die.

Unlike computer chess games, which sometimes ponder their next move for hours on end, *Backgammon 5.0* thinks on its feet, never taking more than a few seconds to decide on a move. Chess games are slow because they look at all the consequences of every legal move; the higher the playing level, the more moves ahead they look. To create the rapid pace and excitement you'd expect in what's basically a gambler's game, *Backgammon 5.0* takes a different approach: As the playing level increases, it uses more advanced strategy in selecting its moves.

If you're looking for a way to improve your play against flesh-and-blood opponents, *Backgammon 5.0* offers you a myriad of options. For instance, if you're not sure what move to make, you can ask the computer to suggest one. After a while, you'll develop a feel for the best way to play various positions. You can also have the computer play against itself so you can observe its offensive and defensive strategies. And if you want to try out special board positions from your favorite strategy book, you can arrange the men on the board any way you want. Other options include changing the skill level during play, exchanging sides with the computer (a nice way to snatch victory from the jaws of defeat), backing up to change a move, automatically playing out the game when the outcome is certain, getting an instant replay, and using external dice. You can save up to sixty programs on disk.

The manual is well written and fun to read. Besides clearly explaining how to use the program, including some tips on strategy, it delves into the game's history. (Did you know Thomas Jefferson dropped a lot of Continentals at backgammon?) If you're caught without the manual, a series of on-screen help menus can pull you through.

Backgammon 5.0 is the first offering for the IBM from Odesta, a company well known to Apple aficionados. Odesta's *Chess 7.0* is considered a classic example of artificial intelligence on a micro; *Backgammon 5.0* is cut from the same mold. Whether you're new to backgammon or a compulsive gammoner, this program will give you hours of fun.

Backgammon 5.0, by Willy Chaplin. Odesta (3186 Doolittle Drive, Northbrook, IL 60062; 800-323-5423). $49.95. ▲

—Terry Datz & F. Lloyd Datz

YOUR SPREADSHEETS CAN LOOK PERFECT AND STILL BE WRONG.

You know how it is.

You spend hours building up a spreadsheet. Inserting data. Changing figures. Adding new formulas suggested by others.

When you're finished, everything looks perfect. But is it?

Are those totals really right?

What formulas produced that bottom line?

Is it out? And if so, *by how much?*

The only way to be sure.

If you use Lotus™ 1-2-3,™ VisiCalc,® or SuperCalc,™ you need The Spreadsheet AUDITOR.

Why? Because only the AUDITOR allows you to quickly and simply print out every formula in your spreadsheet.

Formulas are displayed in a two-dimensional grid that matches the layout of your spreadsheet exactly.

Coordinates are printed. Pages are numbered.

So you can refer to any section instantly — identify changing patterns — and spot errors or omissions, *without* having to move your cursor from cell to cell.

For Lotus users, the AUDITOR even provides a complete listing of all named and special ranges.

Quick, simple, sure.

You don't have to waste time altering column widths. Or worry about clipping important formulas.

The AUDITOR scales columns automatically.

It gives you the option of wrapping long formulas into manageable columns that are easy to read.

A permanent record.

The AUDITOR allows you to document any spreadsheet, and create a permanent record.

It lets you examine the logic behind spreadsheets "handed down" by previous users.

And generates formats that can be followed by other users later.

Other company benefits.

In companies where several people are using spreadsheets, a series of listings can grow to become an application's library.

The AUDITOR makes sure everyone knows what spreadsheets have been developed, so no-one wastes time reinventing the wheel.

A small price to pay.

Best of all the AUDITOR costs just $99.

And that's a small price to pay for spreadsheets you *know* are right.

To get your copy.

Ask your local dealer for the Auditor. If they don't have it yet, call our toll-free order number **800-645-5501**.

(We accept VISA and MasterCard)

Or send a check or money order for $99 to:

Consumers Software Inc.
Dept. B, Ste. 106, 314 East Holly Street
Bellingham, Wa. 98225

Please specify whether you have an IBM/PC or an Apple II or IIe.

For more information:
contact Consumers Software at
(604) 688-4548
(Dealer enquiries welcome.)

™Signifies manufacturers' trademarks.
®Signifies manufacturers' registered trademark.

THE SPREADSHEET AUDITOR™

FOR SPREADSHEETS YOU CAN COUNT ON

marketalk news

Unless otherwise indicated, software listed runs in DOS on machines with either display adapter and requires 64K and at least one disk drive.

Δ **On-Disk Software** announces *RxSet*, a utility program for PC-compatible computers using the Epson RX-80 printer. The program allows the setting of printer control codes and functions through a series of interactive screens. Options that can be set include page length, margins, and line and character spacing. The program provides a current status report on all parameters as they are set and warns of any that are incompatible or may cause problems. Requires DOS 2.0. $15.25.

Δ **Persoft** (2740 Ski Lane, Madison, WI 53713; 608-273-6000) has announced *Smarterm 100*, a full-featured terminal emulation/file transfer package that allows the PC to function as a DEC VT100, VT101, VT102, or VT52 terminal and to transfer ASCII or binary program and data files between the PC and a host computer system. Requires 128K and asynchronous adapter. $149.

Δ **Culverin Corporation** (1440 Yankee Park Place, Dayton, OH 45459; 513-435-2335) offers *Lifestyle Budgeting*, a budget-planning system that incorporates the human aspects of budgeting in addition to the mechanics. It includes the book *Spending Less and Enjoying It More* and does not require detailed record keeping. $49.95.

Δ **Gold Hill Computers** (163 Harvard Street, Cambridge, MA 02139; 617-492-2071) has announced *Golden Common Lisp*, an extensive subset of Common Lisp, the language most used for artificial intelligence programs. Requires DOS 2.0 and 256K. $375.

Δ A macro program that operates within a *1-2-3* worksheet, *Key II*, from **Lighthouse Software Corporation** (Box 15, Hilton Head Island, SC 29938; 803-785-4949), reduces the keystrokes and time required for information management by more than 90 percent. The program consists of six analysis routines and accommodates worksheet files with up to thirty fields and 1,800 records. $189.

Δ **Select Information Systems** (919 Sir Francis Drake Boulevard, Kentfield, CA 94904; 415-459-4003) has announced a word processor in a book. *Select Write* will be packaged like a bestselling novel and is aimed at novice users. $99.

Δ **Micro Architect** (6 Great Pines Avenue, Burlington, MA 01803; 617-273-5658) has brought out *WORD-Jr*, a word processor for the PC*jr* that supports both color and monochrome monitors. It includes a full-screen editor and a text processor and has built-in file/merge capabilities. $59.

Δ **Software Publishing Corporation** (1901 Landings Drive, Mountain View, CA 94043; 415-962-8910) is entering the communications market with *PFS:Access*, an easy-to-use communications package that provides data encryption access information, time-sharing, communication, and home banking services. The package will run on PC*jr*, too. Requires 128K. $95.

Δ **Pearlsoft Division** (Relational Systems International Corporation, 25195 S.W. Parkway, Wilsonville, OR 97070; 503-682-3636) offers *Accounting Pearl*, an integrated accounting package that includes accounts receivable, accounts payable, payroll, and inventory elements. Requires two disk drives, 128K, and printer. $895.

Δ **Johnson Software Company** (420 Pier Avenue, Hermosa Beach, CA 90254; 213-374-6181) has announced *Christmas List*, a personal mailing program that prepares, maintains, and prints a database of labels, tags, three-by-five-inch cards, or similar groups of data. The database allows up to three thousand labels per file. Requires 128K, eighty-column display. $30.

Δ The *Personal Computer Userfest/Chicago*, produced by **Northwest Expositions** (822 Boylston Street, Chestnut Hill, MA 02167; 617-739-2000) is a consolidation of Applefest for Apple Computer users and PC '83 for PC owners. The show will be held Thursday through Sunday, May 3 to 6, 1984, at the O'Hare Exposition Center in Rosemont, Illinois. Admission is $10 for a one-day ticket or $20 for four days.

Δ *Conquest* is a game from **Windmill Software** (2209 Leominster Drive, Burlington, Ontario, Canada L7P 3W8; 416-336-3353) that places the player in a prehistoric era. Represented as a falcon, the player aims to fly higher than his enemy, swoop down, and collide with him. Allows either individual play or two players simultaneously as well as a choice of joystick or keyboard control. Requires color/graphics card. $39.95.

Δ **POSSE** (2828 Forest Lane, Dallas, TX 75234; 214-241-9444) has developed *SWE-SAU*, a polling/sales audit package for the XT that asynchronously polls Sweda L-50 cash registers, pinpoints missing or invalid data, and prints an exception report outlining problems found. Once edited, the program's sales reports are printed. $1,000.

Δ **Strategic Software Systems** (1300 Dove Street, Newport Beach, CA 92660; 714-476-2842) has announced *Bottomline-V*, a corporate financial planning application system for the PC and XT. The system begins with a twelve-month detailed budget. The user then inputs basic data variables, and the system performs more than nine thousand calculations to produce five-year pro formas of profit and loss, balance sheet, changes in financial position, and financial ratio analysis. The program is compatible with most popular spreadsheets. $295.

Δ *Colormax* is a single-board, multidisplay card for the PC and XT from **Micromax** (6868 Nancy Ridge Drive, San Diego, CA 92121; 619-457-3131) that allows simultaneous connection of a monochrome and a color monitor. The board features connectors for composite, RGB, and monochrome displays as well as on-board light pen circuitry. $499.95. With optional parallel interface, $559.95.

Δ *MoneyMaker* is a budgeting program from **Money Tree Software** (760 S.W. Madison Avenue, Corvallis, OR 97333; 503-757-1114) that uses the 10-20-70 budgeting philosophy to help wage earners and professionals set personal financial goals. Requires *VisiCalc*. $139.

Δ **Relational Database Systems** (2741 East Bayshore Road, Palo Alto, CA 94303; 415-424-1300) has announced *Informix*. Originally introduced as a database management system for Unix-based computers, the program's capabilities include query by forms, custom data entry screens, a full-featured report-writing language, and an ad hoc query language. $795.

Δ *Vprint* is a text formatter offered by **CompuView Products** (1955 Pauline Boulevard, Ann Arbor, MI 48103; 313-996-1299). It is designed to be used with all editors and provides full word processing and printing capabilities. $75.

Δ Two new games from **Mirror Images** (504 Broadway, Troy, NY 12180; 518-274-2335). *Freddy Fish* presents a fish-eye view of ocean life. Freddy takes on an aquatic quest to free his captured friends while attempting to evade sharks, barracudas, and the ultimate adversary—man. Playable with keyboard or joystick. Requires color/graphics card. $44.95. Δ *Fleet Sweep* pits players against dozens of different ene-

PC NETWORK

Buy Hardware/Software at Wholesale, And Save On Software Rentals, As A NETWORK™ Member Only!

Save hundreds of dollars when you buy DIRECT from America's Number 1 Computer Buying Service at just 8% above DEALER WHOLESALE PRICES, plus shipping.

Members receive The Personal Computer NETWORK's Giant Catalog featuring over 10,000 products and the lowest prices on the widest selection of computer software and hardware in the nation!

RENT BEFORE YOU BUY—Members are eligible to join The NETWORK's Business and Game Software Rental Libraries for a much smaller fee than other software rental services. And The NETWORK's rental charges are far less — just 20%-25% of the Member WHOLESALE PRICE.

Listed below are just a few of the over 10,000 products available at our EVERYDAY LOW PRICES!

GAMES & EDUCATIONAL SOFTWARE

Please add $1 shipping and handling for each title ordered from below.

	Wholesale Price		Wholesale Price
Infocom *Zork I*	$23.00*	**Spinnaker** *Alphabet Zoo*	$17.00*
Infocom *Zork II*	23.00*	**Spinnaker** *Delta Drawing*	29.00*
Infocom *Zork III*	23.00*	**Spinnaker** *FaceMaker*	20.00*
Infocom *Deadline*	29.00*	**Spinnaker** *Hey Diddle Diddle*	17.00*
Infocom *Witness*	29.00*	**Spinnaker** *Kinder Comp*	17.00*
Lightening *Mastertype*	29.00*	**Spinnaker** *Rhymes & Riddles*	17.00*
Microsoft *Flight Simulator*	29.00*	**Spinnaker** *Snooper Troops #1*	26.00*
Orion *JBird*	22.00*	**Spinnaker** *Snooper Troops #2*	26.00*
Orion *PCMan*	21.00*	**Spinnaker** *Story Machine*	20.00*
Sierra On-Line *Frogger*	21.00*	**Spinnaker** *Most Amazing Thing*	23.00*
Sierra On-Line *Crossfire*	18.00*	**CBS** *Mastering the SAT*	90.00*
Sub logic *Night Mission Pinball*	24.00*	**CBS** *Goren-Bridge Made Easy*	48.00*

BUSINESS SOFTWARE

Please add $2.50 shipping and handling for each title ordered from below.

	Wholesale Price		Wholesale Price
ATI *How To Use MultiMate*	$45.00*	**Monogram** *Dollars & Sense*	$85.00*
ATI *How To Use Microsoft Word*	45.00*	**Oasis** *The Word Plus*	90.00*
Context *Context MBA*	417.00*	**Peter Norton** *Norton Utilities*	48.00*
Conceptual Instr. *Desk Organizer*	150.00*	**Rosesoft** *Prokey Version 3.0*	79.00*
Digital Research *CP/M-86*	33.00*	**Rogue** *Micro SPF*	76.00*
Funk Software *Sideways*	36.00*	**Software Publishing** *PFS: File*	82.00*
Lotus Development *Lotus 1-2-3*	CALL	**Software Publishing** *PFS: Graph*	82.00*
MDBS *Knowledge Man*	275.00*	**Software Publishing** *PFS: Report*	68.00*
Micro Rim *R Base 4000*	290.00*	**Software Publishing** *PFS: Write*	82.00*
Microsoft *C Compiler*	300.00*	**Softword Systems** *MultiMate*	260.00*
Microsoft *Pascal Compiler*	210.00*	**Sorcim** *Supercalc III*	217.00*
Microsoft *Word with Mouse*	260.00*	**SPI** *Open Access*	350.00*

HARDWARE

Please add appropriate shipping and handling charges (at right in parentheses) to each item ordered from below.

	Wholesale Price		Wholesale Price
Amdek *Video 300*	$115.00* (3.00)	**NEC** *Spinwriter 3550 35 CPS letter-quality printer*	$1,604.00* (32.00)
Amdek *Video 300A*	130.00* (2.60)	**NEC** *2050 Spinwriter 20 CPS*	865.00* (17.50)
Amdek *Video 310A*	140.00* (2.60)	**Okidata** *ML92P*	CALL
Amdek *MAI multiple 4-in-1 display board*	415.00* (2.50)	**Okidata** *ML93P*	CALL
Anchor *Mark XII, lowest priced Hayes compatible 1200BPS external modem*	230.00* (5.00)	**Okidata** *IBM to Okidata Cable*	20.75* (1.00)
		Persyst *Monochrome Card*	205.50* (2.50)
Apparat *512K RAM Card w/0K*	92.00* (2.50)	**Paradise (UNIV. RSCH.)** *Multi-display card*	345.00* (2.50)
AST *Six-Pack Plus-0K*	210.00* (2.50)	**Princeton Graphics** *HX12*	CALL
AST *Megaplus II-0K*	210.00* (2.50)	**Quadram** *Quad Color I*	191.00* (2.50)
AST *I/O Plus II*	99.00* (2.50)	**Qume** *Half Height DS DD Disk Drives (IBM Portable equiv.)*	350.00* (7.00)
AST *Any Port Option (MG II, SPK, I/O+)*	30.00* (1.00)	**Qume** *Sprint II 40 CPS*	1,220.00* (24.40)
Epson *FX-80*	CALL	**Qume** *Sprint II 55 CPS*	1,359.00* (27.18)
Epson *FX-100*	CALL	**Qume** *IBM IFCE w/cable*	72.00* (2.50)
Epson *RX-80*	CALL	**Qume** *Daisy Wheels*	6.00*
Epson *Epson to IBM printer cable*	21.00* (1.00)	**Qume** *Ribbons*	6.00*
Hayes *Smartmodem 300*	183.00* (3.60)	**STB** *Graphix Plus*	297.00* (2.50)
Hayes *Smartmodem 1200B*	369.00* (2.50)	**STB** *Clock Option*	30.00* (1.00)
Hercules *Graphics Card*	320.00* (2.50)	**Tandon** *TM100-2 (DS/DD IBM equiv.)*	200.00* (4.00)
IDS *Prism 132C (color, w/all 4 options)*	1,390.00* (30.00)	**Tallgrass** *20Mb Hardfile (complete system)*	2,310.10* (46.20)
IDS *Prism to IBM cable*	20.00* (1.00)	**Teva** *Tava PC IBM Compatible*	1,650.00* (3.00)
Kraft *Analog joystick*	29.97* (2.50)	**Transtar** *120P, 15 CPS letter-quality printer*	387.00* (6.35)
Maynard *Floppy Disk Controller*	155.00* (2.50)		
Maynard *WS1 10Mb Internal Hard Disk/Multifunction*	855.00* (17.60)	**USI** *Pi-3 Monitor (12" amber)*	132.00* (3.00)
Maynard *WS2 10Mb Internal Hard Disk/Floppy Controller*	960.00* (19.20)		

MEMORY

	Wholesale Price		Wholesale Price
Maynard *WS3 10Mb Internal Hard Disk/Sandstar Memory w/0K*	960.00* (19.20)	**64K Memory Set** *for IBM PC, compatibles, & memory boards*	$45.90* (1.00)
		Installation of Memory on board	10.00
NEC *Notebook Computer*	559.00* (11.18)	**Brand Name** *DS DD diskettes lifetime guarantee (box of 10)*	22.50* (2.00)

*PC NETWORK Members pay just 8% above the wholesale price, plus shipping. All Prices Cash. On credit card orders there is a 3% service charge.

Low prices, fast home delivery and two software rental libraries are only the beginning! The NETWORK is your source for everything from memory chips to mainframes—and it's all just 8% above wholesale, plus shipping.

HARDWARE

Monitors (color and monochrome)
Printers
Complete Systems
Disk Drives (full/half height, add-on/add-in)
Multi-Function Boards
Graphics Boards
Modems
Local Area Networks
Memory Chips (all speeds available)
S-100 Components

SOFTWARE (rent or buy!)

Available in most major system formats.

Business	**CPM-80**
Recreational	**CPM-86**
Compilers	**MS-DOS**
Word Processors	**PC-DOS**
Utilities	**Apple-DOS**
Scientific	**TRS-DOS**
Data Bases	
Educational	
Graphics	**And More!**

SUPPLIES & ACCESSORIES

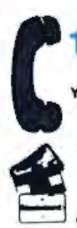
Blank Diskettes (all formats)	**Print Wheels**
	Cables
Paper Stock	**And More!**
Ribbons	

Choose hardware and software from hundreds of manufacturers, including:

- Altos
- Amdek
- Apparat
- AST
- Ashton-Tate
- Atari
- CDC
- Coleco
- Columbia
- Commodore
- Corona
- Cromemco
- DEC
- Digital Research
- Eagle
- Epson
- Franklin
- Hayes
- IBM
- IUS
- Lotus
- MicroPro
- Microsoft
- Morrow
- Motorola
- NEC
- Northstar
- Okidata
- Peach Tree
- Princeton
- Quadram
- Sanyo
- Sierra On-Line
- Software Publishing
- Softword
- Tandon
- Tecmar
- Texas Instruments
- Toshiba
- Visicorp
- Xerox
- Zenith

1. **REAL BUYING CLOUT**—Buy at just 8% above DEALER WHOLESALE PRICES, plus shipping.
2. **SHOP-AT-HOME CATALOG**
3. **KNOWLEDGEABLE CONSULTANTS**
4. **INSURED FAST HOME DELIVERY**
5. **OPTIONAL BUSINESS SOFTWARE RENTAL LIBRARY—Rent business software at just 20%-25% of The NETWORK's low prices** for a 7-day period (plus a 3-day grace period for return shipping). **100% of your rental fee applies towards purchase.**
6. **OPTIONAL GAME SOFTWARE RENTAL LIBRARY**—All the same conditions apply as for benefit five.
7. **SPECIAL SAVINGS BULLETINS**
8. **DISCOUNT BOOK LIBRARY**
9. **MEMBERSHIP GUARANTEE**
10. **PRODUCT GUARANTEE**

When ordering, be sure to provide us with your type of computer, amount of memory, number and type of disk drives and type of monitor (color or monochrome).



CALL TOLL FREE
1-800-621-S-A-V-E
In Illinois call (312) 280-0002

Your Membership Validation Number: S700
You can validate your membership number and, if you wish, place your first money-saving order over the phone by using your VISA, MASTERCARD or AMERICAN EXPRESS. Our knowledgeable service consultants are on duty Mon-Fri 8 AM to 7 PM, Sat 9 AM to 5 PM CST.

Call now...Join the NETWORK and start saving today!

PERSONAL COMPUTER NETWORK
320 West Ohio
Chicago, Illinois 60610

All items subject to availability.
prices subject to change without notice

THE NETWORK · MEMBERSHIP APPLICATION

YES! Please enroll me as a member of The NETWORK and rush my catalog featuring thousands of computer hardware and software products, all at just 8% above DEALER WHOLESALE PRICES. I will also receive all the other exclusive, money-saving services available to Members. I am under no obligation to buy anything. My complete satisfaction is guaranteed.

Please check (✔) all boxes that apply:

Basic Membership

- ☐ One-year membership for $8
- ☐ Two-year membership for $15 (SAVE $1)
- ☐ Business Software Rental Library for $30 add'l. per year—members only
- ☐ Games Software Rental Library for $10 add'l. per year—members only

Special V.I.P. Membership 700

Includes advance notification of special limited-quantity merchandise.

- ☐ One-year membership for $15
- ☐ Two-year membership for $25 (SAVE $5)
- ☐ BOTH Business and Game Software Rental Libraries for $35 add'l. per year—Extended 10-day rental to V.I.P. members only

☐ Bill my credit card: ☐ VISA
☐ MasterCard ☐ American Express Exp. Date ______ mo. year

Account Number: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

☐ Check or money order enclosed for $ ______

Name ______

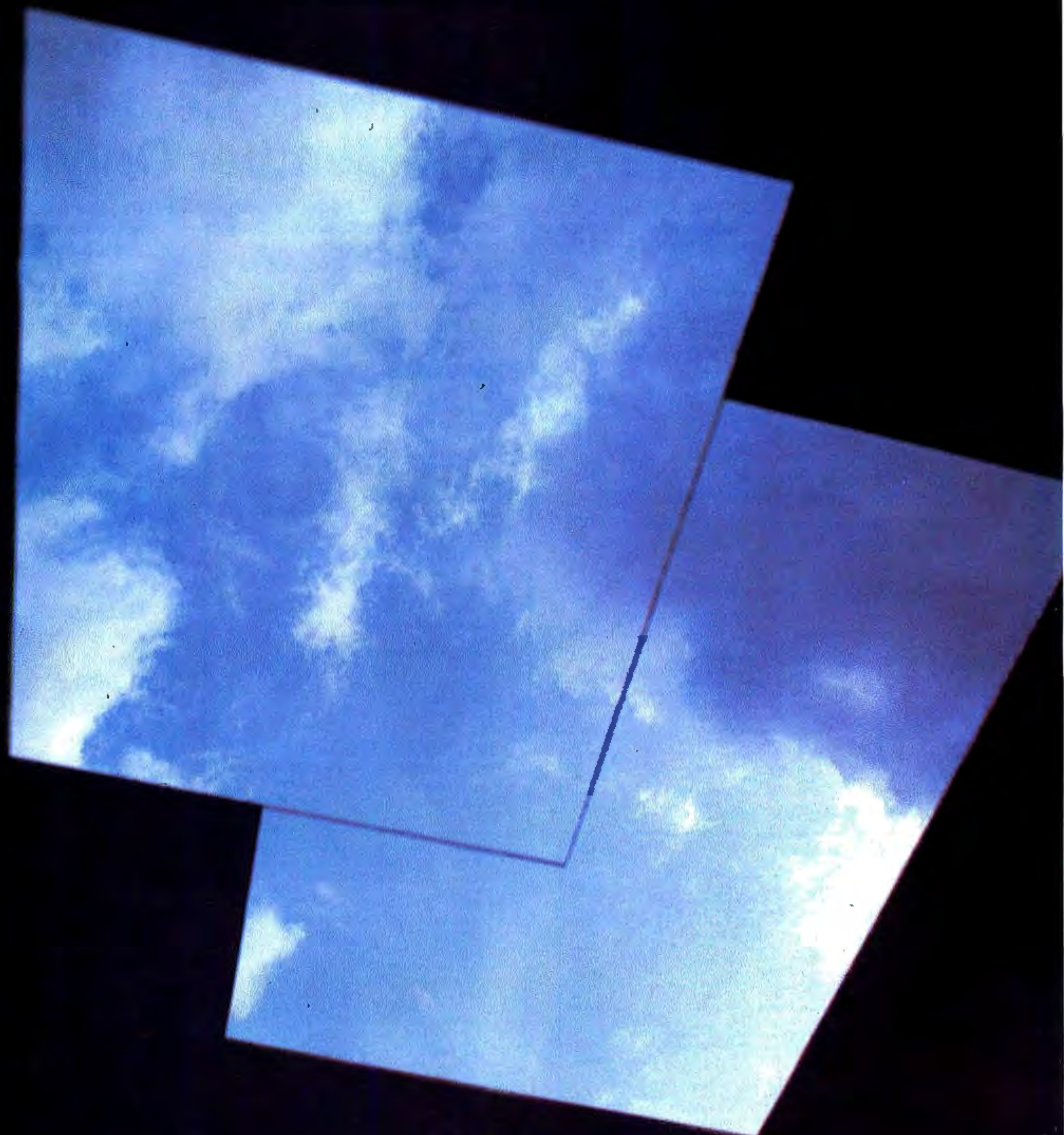
Address ______ Apt. No. ______

City ______ State ______ ZIP ______

Telephone (______) ______

My computer(s) is: ☐ IBM PC ☐ Apple II ☐ TRS-80 ☐ Atari ☐ Commodore Other ______

Signature ✔ ______
(Signature required to validate membership)



mies in wave after wave of attacking fleets. Each enemy flies in a different attack pattern and uses its own weaponry. It is the player's job to overcome each enemy and eliminate the invading fleet. Can be played with keyboard or joystick. Requires color/graphics card. $39.95.

Δ *Software Resources* from **Associated Technology** (Route 2, Box 448, Estill Springs, TN 37330; 615-967-9159) is a tool for estimating schedules and other resources for software development projects. The program aids software managers in estimating staff level, cost, milestones, and schedule length. $175.

Δ Several products from **MAI/Basic Four Business Products Corporation** (601 San Pedro N.E., Albuquerque, NM 87108; 505-266-5811). An integrated accounting series includes four currently available packages: *General Ledger, Accounts Payable, Accounts Receivable,* and *Inventory.* Seven additional programs are slated for release later in the year. Requires 128K. $395 to $595. Δ An enhanced business Basic language, *BB/M* runs under MS-DOS and is designed for both first-time users and professional software developers. $495.

Δ **MicroPro International** (33 San Pablo Avenue, San Rafael, CA 94903; 415-499-1200) is shipping an enhanced version of its spreadsheet software, *CalcStar.* The new version offers extended cell capacity and color implementation. Requires 128K. $195. Update for current users, $60.

Δ **Software Dynamics** (Box 247, Dunedin, FL 34296-0247; 813-733-8784) has announced a series of communication emulators for the PC and XT. *SDI3780* emulates every feature and function of the IBM 3780 RJE workstation with the exception of the home mode and on-line test mode. *SDI2780* emulates every feature and function of the IBM 2780 RJE workstation with the exception of six-bit transcode, synchronous clock feature, and off-line operation. Both require a binary synchronous communications adapter board. $599 each.

Δ An information service for professional investors, *Micro/Scan* from **Isys Corporation** (50 Church Street, Cambridge, MA 02238; 617-491-6221) features large-scale databases on disks, updated daily, with accompanying software. Data can be transferred directly to *VisiCalc* or *1-2-3* files. From $6,250 to $8,500 per year.

Δ *SPSS/PC* is a data management and reporting package from **SPSS** (444 North Michigan Avenue, Chicago, IL 60611; 312-329-2400) that offers users access to the statistical procedures most often applied to solve business and research problems, including univariate statistics, cross tabulation, correlations, multiple regression, analysis of variance, nonparametric tests, factor analysis, and contour and scatter plot procedures. Requires 320K, hard disk, and 8087 coprocessor. $795.

Δ **Scenic Computer Systems Corporation**, 14852 N.E. Thirty-First Circle, Redmond, Washington 98052; 206-885-5500) has introduced *ScenicWriter,* a document composition system that produces output that can be used as camera-ready copy by writers and publishers. The program includes a forty-thousand-word spelling checker. Requires p-System. $575.

Δ Up to nine tasks can be run concurrently with *Tascmaster* from **Profit Systems** (Box 1039, Berkley, MI 48072; 313-559-0444). The program divides system memory under user control into independent sections. $475.

Δ **cum laude Software** (10221 Slater Avenue, Fountain Valley, CA 92708; 714-964-4075) offers *Barney O'Blarney's Magic Spells,* an educational game designed to improve a child's spelling and math-reasoning ability. The word list supplied covers grades 1 through 7, and other words can be added. Requires 128K and color/graphics card. A version for schools that tracks thirty-five students is also available. $33.33.

Δ *DoubleDOS,* an operating system enhancer that lets the PC and XT accomplish two functions simultaneously, is available from **Softlogic Solutions** (530 Chestnut Street, Manchester, NH 03101; 800-272-9900). The program divides a system's memory into two areas and works with both the monochrome and graphics displays either together or separately. The fifteen-character keyboard buffer is extended to two eighty-character buffers. Requires 128K, but at least 192K recommended. $299.

Δ **Realia** (10 South Riverside Plaza, Chicago, IL 60606; 312-346-0642) has announced *Realia Cobol,* a compiler that allows large IBM VS Cobol programs to be compiled and tested on the PC and XT using a floppy or hard disk. Typical compilation speed is 700 to 1,500 lines per minute on floppy and 1,200 to 2,600 on hard disk, depending on compiler options. $995.

Δ **IBM** (Entry Systems Division, Box 2989, Delray Beach, FL 33444; 800-426-2700) has announced its *IBM Personal Computer Software Support Center,* a subscription service that offers technical assistance for fifteen selected IBM software products. This is a pilot program offered within a limited geographic area (Alabama, Arkansas, California, Delaware, Florida, Georgia, Indiana, Kentucky, Louisiana, Maryland, Mississippi, North Carolina, South Carolina, Tennessee, Virginia, West Virginia, and the District of Columbia) and costs $380 for seven single-topic phone calls as well as a monthly software newsletter.

Δ **International Microcomputer Software** (633 Fifth Avenue, San Rafael, CA 94901; 415-454-7101) has published an investment software package that allows individual investors, professional portfolio managers, and financial planners to calculate how much of their return is due to decision making and how much is due to luck. Called *Investment Manager,* the program reports investment performance and risk and return and then weighs those factors in comparison with alternative investments. Requires 128K. $275.

Δ **Marnoble Software** (6100 Hillcroft, Houston, TX 77081; 713-981-6053) has released *Taxpertise,* an income tax package designed to run on *1-2-3.* Requires 256K. $250.

Δ *p-Comm* is a professional communications package from **ErgoSoft** (7122 Cather Court, San Diego, CA 92122) for users of the UCSD p-System. It is designed to complement remote computer services such as The Source and CompuServe, and supports a wide variety of file-transfer protocols, including the Christensen, or XMODEM, protocol. $95.

Δ **STSC** (2115 East Jefferson Street, Rockville, MD 20852; 301-984-5000) has announced release 3.0 of *APL *PLUS/PC System.* New features include a full-screen editor, scrolling display screens, support for DOS 2.0, graphics primitives, an on-line help facility, and English-like keywords. Requires 196K. $595.

Δ *Mailer,* a standalone mailing list program for U.S. and Canadian mail, is available from **Maurizi Associates** (1344 Fitch Way, Sacramento, CA 95825; 916-486-2993). The database is already designed, as are the formats for printing labels or addressing envelopes. It can read in existing address files. $150.

Δ **Office Solutions** (5708 Odana Road, Madison, WI 53719; 608-274-5047) has released a new version of *OfficeWriter,* a Wang-like word processor designed with the novice in mind. $325.

ΔTwo new financial software packages are available from **MCS Software** (2715 East 3300 South, Salt Lake City, UT 84109; 801-486-8746). *BancLoan* is a flexible loan calculation and document preparation package with the capability to compute nearly all types of consumer and commercial loans, produce all loan disclosure information required for federal compliance, and prepare any documents needed. It also allows for loan computation recall and recalculation, instant amortization to zero, and table maintenance procedures. Requires 128K. $1,295. Δ *StructureLease* is a package that will assist leasing firms in structuring simple investor leases. The program allows the lease company to avoid having to use costly time-sharing fees. Requires 128K. $1,995.

Δ **Summa Software Corporation** (Box 2046, Beaverton, OR 97075; 503-644-3212) introduces *Winning on Wall Street,* a stock-market decision support system for the PC and XT. It includes a comprehensive database, forecasting/technical analysis, and portfolio management

system. $700. The product also can be purchased as three separate sections: *Trader's Data Manager*, $200; *Trader's Forecaster*, $250; and *Trader's Accountant*, $350.

Δ Three products that can be integrated with one another are available from **Innovative Software** (9300 West 110th Street, Overland Park, KS 66210; 913-383-1089). *The Smart Data Manager* can be used for creating various filing systems, generating reports, forms, and files, and creating mailing labels and invoices. $595. Δ *The Smart Spread Sheet with Graphics* handles large amounts of data in many formats while executing many different commands. The largest spreadsheet can be 999 columns by 9,999 rows. $595. Δ *The Smart Word Processor* allows multiple documents or parts of documents to be seen on-screen through windowing. A document can include characters of different fonts, sizes, and colors. $475.

Δ *Practical Accountant* from **Softlink** (3255-2 Scott Boulevard, Santa Clara, CA 95051; 408-988-8011) is a single-entry accounting package that handles receipts, deposits, credit cards, accounts payable, accounts receivable, and simple payroll. Requires 128K (DOS 2.0, 192K). $149.95.

Δ *Mini-Ledger* is an accounting system from **Paradigm Consultants** (39243 Liberty Street, Fremont, CA 94538; 415-796-0543) that is designed for small businesses with up to twenty-five employees. Requires CP/M-86. $150.

Δ An as-yet-unnamed *electronic disk* is available from **Distributed Logic Corporation** (12800 Garden Grove Boulevard, Garden Grove, CA 92643; 714-534-8950). The disk plugs into any I/O slot and is fifty times faster than a floppy. The unit has an independent power supply and no moving parts. $1,595.

Δ *3 Plus*, from **The Phoenix Consulting Group** (Box 191016, Dallas, TX 75219; 214-263-6455), runs with *1-2-3* and provides a series of templates as well as reporting capabilities for actual and projected data. Requires 256K. $289.

Δ *PeopleTrak*, an association management package for the XT, has been introduced by **Noesis Computing Company** (615 Third Street, San Francisco, CA 94107; 415-495-7440). The program manages membership information and allows users to generate their own reports with minimal training. $1,995.

Δ **Software Solutions** (305 Bic Drive, Milford, CT 06460; 212-986-6668) has introduced a relational database management system, *Dataease*, that allows users to develop applications without programming knowledge. Requires DOS 2.0. $595.

Δ **Visualtek** (1610 Twenty-Sixth Street, Santa Monica, CA 90404; 213-829-6841) has announced the *Model DP-11*, a peripheral display that enlarges letters up to sixteen times their normal size. The device is transparent and requires no special software. $2,470.

Δ **Individual Software** (1163-I Chess Drive, Foster City, CA 94404; 415-341-6116) has announced three software tutorials for IBM personal computers, including PC*jr*. *Typing Instructor* provides instructional lessons with varied choices of instructional methods. $49.95. Δ *Professor Pixel* is an interactive training program that introduces the beginner to creating graphics, sounds, color, and animation using BASIC. $59.95.

Δ The *Bubble Drive*, a bubble memory board, is available from **HI-COMP Computer Corporation** (5016 148th Avenue N.E., Redmond, WA 98052; 206-881-6030). It provides either 256K or 512K of nonvolatile, high-speed mass storage on a single card that plugs into any I/O slot. 256K, $995. 512K, $1,495.

Δ **Coefficient Systems** (611 Broadway, New York, NY 10012; 212-777-6707) has announced *VTERM II*, emulation software that combines full emulation of the DEC VT100, VT101, VT102, and VT52 terminals with new features not yet available in any terminal. The program lets users hook their PCs up to any DEC mainframe and run all the host software unchanged. $160.

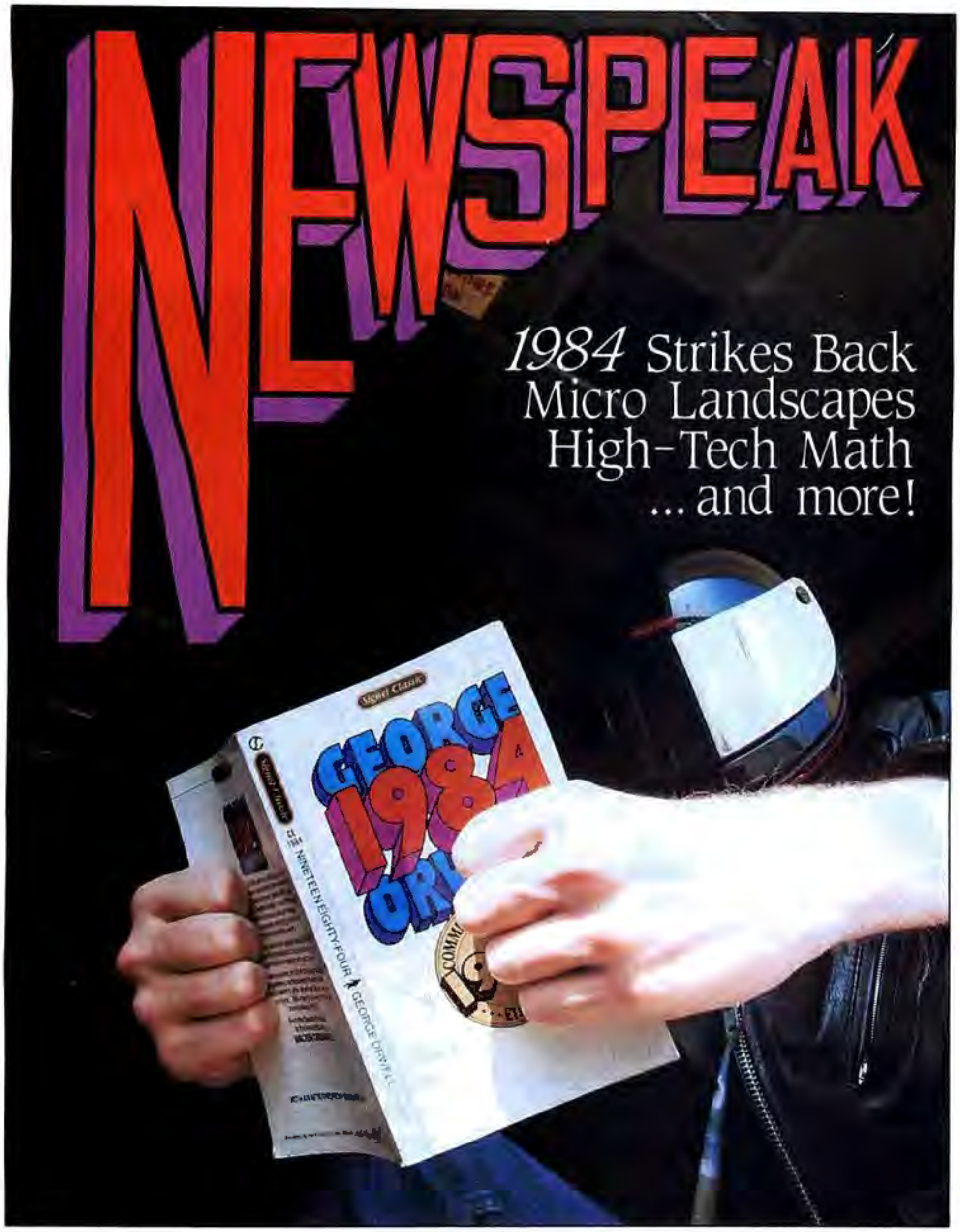
Δ Enhanced versions of two file management systems have been released by **Softcraft** (Box 9802 #590, Austin, TX 78766; 512-346-8380). *Btrieve 3.0* (for standalone use) and *Btrieve/N 3.0* (for use in a local-area network) include enhancements such as new file characteristics, a sorting capability, more efficient use of disk space, and high-speed file handling options. *Btrieve*, $245. *Btrieve/N*, $595.

Δ **Persyst** (15801 Rockfield Boulevard, Irvine, CA 92714; 714-859-8871) has announced a color display adapter that offers high-resolution character display and compatibility with IBM color/graphics adapter software. The *BoB Display Adapter* displays an eight-by-twelve-dot character in a ten-by-sixteen-dot grid. $425.

Δ **Continental Software** (11223 South Hindry Avenue, Los Angeles, CA 90045; 213-417-8031) introduces two programs. *Property Management* performs all bookkeeping functions for any residential or commercial property with as many as one thousand renters or lessors, depending on the hardware used. Financial records include income and expense summaries, and individual tenant records. $495. Δ *Ultrafile* is an integrated file manager that accommodates databases of up to 32,000 records, 1,000 characters per record, fifty data items per record, and 100 characters per data item. Reporting and graphic functions are included. A color/display card is required for the latter. Requires 128K and two double-sided disk drives, one double-sided drive, or a hard disk. $195.

Δ *Snapshot*, from **Computer Headquarters** (333 Peters Street S.W., Atlanta, GA 30313; 404-577-1111), allows the operator to create an instant picture of almost any graphics display on a wide variety of printers that have dot-addressable graphics. $65.

Δ *PC Master*, a self-learning software package from **CourseWare** (10075 Carroll Canyon Road, San Diego, CA 92131; 619-578-1700), has been enhanced to include sections for the PC*jr*. The interactive package helps computer novices teach themselves to operate their computers. $79.50. ▲



ORWELL AND BIG BROTHER ARE BIG BUSINESS IN 1984

Is it a warning or a prophecy? Is it a truly insightful intellectual exercise or is it, as Frederic Mullally, a contemporary of George Orwell's, recently put it, a "bogus literary time bomb defused from its inception by the myopia of the author's political vision." You call it.

Now thirty-six years old, George Orwell's *Nineteen Eighty-four* (the author preferred the title written out) is probably as controversial and widely read as ever. A stark, moody novel, it is a vision of a negative utopia—with its thought police, antisex organizations, junior spies, all-seeing telescreens, and novel-writing machines. Currently, it's one of the top five bestselling paperbacks—right up there with *Lonesome Gods, Megatrends, Ascent into Hell, Utterly Gross Jokes,* and *The Michael Jackson Story.*

From Apple's "1984" TV commercial touting Macintosh to Van Halen's latest heavy-metal opus *1984*, the list of homages, condemnations, rip-offs, fraternal nods, and (by far the largest group) shots in the dark somehow connected with Orwell and his last novel is already embarrassingly long. And that's just what has been said, written, filmed, advertised, recorded, and produced in the first two months of this year.

Few literary works have caused such a ruckus. Thirty years ago, Signet's twenty-

GOTO page 177, column 3

INDUSTRY TRIES TO FIND ITSELF IN 1984 SEASON

Spring is the season of Oscars, Opening Days, and the Fortune 500. Spring is the time for planting, flooding, cleaning, and putting clocks ahead.

This spring, the computer industry is trying to find itself. As baseball managers experiment with different starting line-ups and team rosters, industry executives are looking for the right combinations of talent and experience to produce winning teams (that in turn produce winning products). At the same time, fears of impending disaster are being felt throughout the industry. Many manufacturers are struggling. Software publishers are either desperately spending huge amounts of money or desperately peering into crystal balls.

Deep down, everybody in the industry still believes that micros are the hottest thing since color TV. And everybody wants a piece of the predicted millions that will be made later this decade. The problem is surviving until that pie is cooked and ready to be divvied up.

What follows are facts, rumors, updates, obituaries, and general scuttlebutt about the computer industry—a kind of *"Computers Illustrated* Preview of the Coming Season." Make your bets and buy an extra-large popcorn: The game has just begun.

Facts: IBM plans to sell $1.2 billion worth of personal computers in 1984 (the company sold $700 million worth in 1983); Apple would like to sell a million Macintoshes; AT&T is hungry and on the move, threatening to enter the personal computer market sometime this year (Ma Bell posted a $4.87-billion loss in the last three quarters of 1983, more than four times the previous record loss).

Fact: On February 16, 1984, IBM announced the IBM Portable Personal Computer—an 8088-based machine with 256K and one disk drive. The machine is bound to cause some headaches for Houston-based Compaq Computer and KayPro of Solana Beach, California.

Fact: Earlier this year, AT&T bought a 25 percent interest in Olivetti, the Italian manufacturer of office equipment, for about $260 million. AT&T also is engaged in cooperative ventures with Philips of Holland, Wang Laboratories, Digital Research, and Convergent Technologies.

GOTO page 181, column 1

Cash Award for Math Programs Is Announced

A $100,000 prize donated by the Fredkin Foundation of Cambridge, Massachusetts, awaits anyone who can design a computer program capable of making a significant mathematical discovery. The prize will be awarded "for a mathematical work of distinction in which some of the pivotal ideas have been found automatically by a computer program in which they were not initially implicit," according to a statement from a committee of mathematicians and computer scientists who are defining the rules for the competition.

Sound easy? Don't bet on it. The creators of one of the most sophisticated programs to date, *Bacon*, believe it will be ten to twenty years before anyone wins the prize.

One difficulty facing programmers is that a candidate program must come up with an original theorem and proof without "coaching" from humans. On its own, the program must discover a theorem that other mathematicians agree is a major new result.

Currently, more than a few research groups are developing "automated theorem-proving" programs. Although great progress has been made with these programs, they still lag behind human capabilities. Programs like *AM* (automated mathematics), created by Douglas B. Lenat at Stanford, and *Aura*, developed at the Argonne National Laboratory in Illinois, are formulating new mathematical ideas and solving previously unsolved problems in advanced mathematics. *Bacon*, created by Herbert A. Simon and Patrick Langley of Carnegie-Mellon University, can generate conjectures from patterns and relationships it discovers within data; the program has already rediscovered Kepler's laws of planetary motion and Snell's law for the refraction of light.

It is hoped that the $100,000 prize (which may double or triple in size because of accumulated interest, depending on how soon someone wins it) will encourage more mathematicians to use computers in mathematical research.

Woodrow W. Bledsoe of the math department at the University of Texas in Austin is chairman of the rules committee. A computer enthusiast, Bledsoe wants to see computers used more in mathematical research. "We want to see if we can make computers act like humans. We want them to be able to reason," he says.

"I think the winner will be a person who is knowledgeable in both computer science and mathematics," says Bledsoe. "And it probably won't be someone we've never heard of." The full criteria for winning the prize will be published soon, says Bledsoe, adding that he doesn't expect to receive any serious candidates for quite a while.

The committee is also considering the idea of an additional prize for a computer program that can take a tough mathematics examination and do better than its human competitors. The Fredkin Foundation has a $100,000 prize ready for the first computer chess program to become the world chess champion. (Recently the Foundation awarded $5,000 to Kenneth Thompson and Joseph Condon of AT&T Bell Laboratories for *Belle*, the first computer system to achieve a Master rating in tournament play.)

The Fredkin Foundation was established by Edward Fredkin, a professor at the Massachusetts Institute of Technology. DH, MS

Orwell

continued from page 175

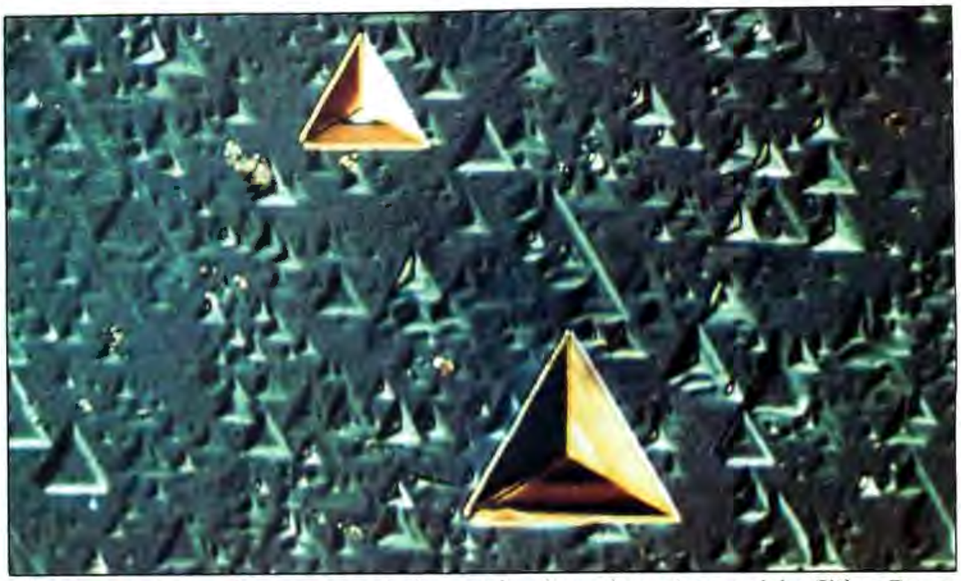
five-cent paperback edition of *Nineteen Eighty-four* asked in bold letters on the back cover, "Which one will YOU be in the year 1984?"—proletarian, police guard, or Party member. "There won't be much choice, of course, if this book's predictions turn out to be true." The blurb typifies the attitude toward the book in the fifties. Anti-Communist sentiment was at its peak, and the ruling Party in *Nineteen Eighty-four*, and its figurehead Big Brother, were considered synonymous with Stalinism and the U.S.S.R.

1984 has been a worry nagging the Western conscience for the last three decades. Ironically, Orwell originally planned to call the book *The Last Man on Earth* and eventually felt that he had ruined the original good idea for the story. But Orwell (his real name was Eric Blair) was always pessimistic about his own writing.

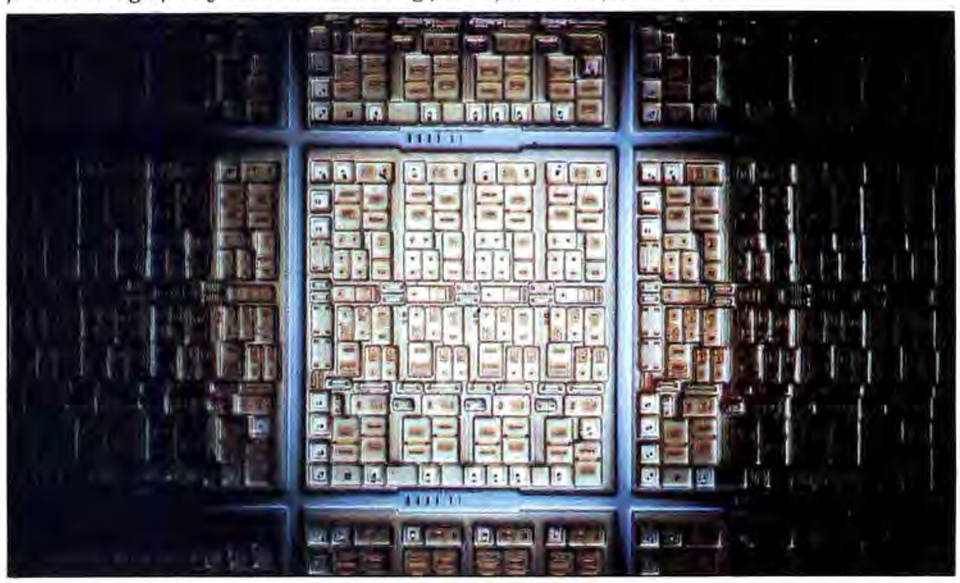
Now, in the dreaded year itself, the book is perceived by many as a long-term warning. Some see the computer revolution, with its promise/threat of such things as two-way videotex and artificial intelligence, as reaching close to Orwell's totalitarian vision. Others see the current massacre of the English language in advertising and politics as evidence that Newspeak—the official language/thought process of the Party in *Nineteen Eighty-four*—is alive and well in the world. And others, like Mullally, say Orwell was a bitter, dying man who botched it.

GOTO page 179, column 1

High-Tech Photography Exhibit Tours US



Above: vaporization pits on arsenic single-crystal surface, photomicrograph by Clifton Draper, enhanced with Normarski differential interference-contrast; below, gated crosspoint diodes, photomacrograph by Charles Lewis using fiber optic- and epi-illumination.



A fascinating exhibit of forty-seven photographs, called "MicroScapes: The Hidden Art of High Technology," is currently making a two-year multicity tour of the United States. Mounted by AT&T Technologies, the exhibit represents some of the current processes used by AT&T in the research, development, and production of advanced communications systems.

The large-format color and black-and-white photos in the exhibit employ such advanced photographic techniques as photomacrography, photomicrography, interferometry, thermography, and light polarization. These state-of-the-art techniques—some of the photos were shot at exposure speeds of 1/720,000th of a second and with magnifications of 67,000 times the actual image size—reveal eerie and colorful landscapes that are hidden from the naked eye.

High-technology manufacturing processes such as electrode deposition of gold and copper, silicon irradiation by laser, plasma etching of silicon wafers, and the joining of materials by explosive bonding are dramatically pictured. The extremely close-up photos of microprocessor chips, glass fibers, crystals, and magnetic bubbles begin to resemble fine art.

Photomacrography is a process by which moderately magnified pictures—generally two to twenty-five times original size—of small subjects are made. Optical photomicrography, through the use of a compound microscope, greatly magnifies minute subjects.

Two electron microscope processes are represented in "MicroScapes." In scanning electron photomicrography a narrow beam of electrons scans the exterior surface of an object to produce an image that appears dimensional. In transmission electron micrography an electron beam penetrates a thin sample of an object, projecting a shadow image that is recorded on photographic film.

Interferometry is a technique by which to visualize invisible density variations in a disturbed medium. It involves splitting a light source into two beams, then recombining them on film.

Thermography is a technique for imaging variations in the amount of infrared radiation, or heat, emitted by a subject. Light polarization is used to differentiate between portions of subjects of varying thickness or optical density.

In addition to the photographs, "MicroScapes" features a number of freestanding, in-

teractive displays—including a one-half-inch-square Western Electric 256K RAM chip containing over six hundred thousand individual components.

"MicroScapes" finishes up a week-and-a-half stint at Baltimore's Maryland Science Center on April 1. The show then moves to Indianapolis and the Children's Museum, where it runs April 21 through June 3. Four more cities are currently on the docket: Fort Lauderdale, Florida, June 23 through August 5; Oklahoma City, August 25 through October 7; Atlanta, November 3 through December 16; and Reading, Pennsylvania, May 11 through June 23, 1985.

For more information contact the Association of Science-Technology Centers in Washington, D.C.

Orwell

continued from page 177

Confused as everybody seems to be about it, *Nineteen Eighty-four* is in any case a hot item. There are still eight months left in the year, and it's a sure bet that we're in for more Orwellian revelry.

Herewith, then, is a brief rundown of recent events, new and forthcoming publications, and future happenings that are in some way related to Orwell. If anyone invites you to an "April 4 Party" or to join the Urban Anti-Sex League, at least you'll be prepared.

In addition to writing the preface to Signet's "Commemorative" edition of *Nineteen Eighty-four,* Walter Cronkite narrated "1984 Revisited," a sixty-minute CBS documentary aired last June. In the preface to the new paperback edition, Cronkite calls *Nineteen Eighty-four* "a novelistic essay on power" and "an anguished lament." He identifies Big Brother with Stalin, Hitler, and Khomeini and identifies Newspeak with the "dehumanizing babble of bureaucracies and computer programs." Cronkite's conclusion? The book is a warning, and "1984 may not arrive on time, but there's always 1985."

National Public Radio is offering "Will Next Year Be '1984'?"—a thirty-minute cassette of a four-part series comparing Orwell's *Nineteen Eighty-four* with the real 1984.

A seventeen-volume edition of Orwell's complete works will be published this year in England and this country. In addition, the BBC is brewing a whole stew's worth of tributes, dramatizations, and documentaries about Orwell and his works. Orwell worked for years as a BBC broadcaster, before and during World War II. Fittingly, a wax figure of Orwell was installed at Madame Tussaud's in London this past December.

Production has begun on a new film version of *Nineteen Eighty-four,* directed by Michael Radford (*Another Time, Another Place*). The original film adaptation of the book was released in 1956, with Edmund O'Brien portraying the protagonist, Winston Smith. Three years ago, Marvin Rosenblum, a Chicago lawyer, bought the film rights from Orwell's widow. Apparently somewhat obsessed with making the movie, Rosenblum spent a long time trying to get directors Hal Ashby, Milos Forman, and Francis Coppola interested in the project. Now Rosenblum is in partnership with British producer Simon Perry and Virgin Pictures, which will provide financing.

This year, Jura—an island in the Hebrides, west of the Scottish coast—is expecting increased tourism because Orwell lived there while writing *Nineteen Eighty-four.* Orwell's island residence, a dismal old manor called Barnhill, is still standing. Jura is usually portrayed as a bleak, chilly, damp place; even so, a minor horde is likely to descend there in search of *Nineteen Eighty-four's* roots.

Two Michigan State graduates have come up with the "1984 Calendar," billed as a day-by-day history of the increasing erosion of civil liberties in the U.S. It features black-and-white photographs of U.S. government buildings (IRS, FBI, and the Bureau of Indian Affairs) and of police riot squads and jail cells. Each date is annotated with one or more reminders of the loss of freedom. Example: On August 1, 1973, the *Washington Post* reported a private investigation launched by the Nixon White House on the Smothers Brothers.

Much has been written in the editorial sections of newspapers about Orwell and his work, and at least two books directly related to *Nineteen Eighty-four* have been published so far this year.

1984 and Beyond, by Nigel Calder (Viking), is written in a bizarre, question-and-answer format. O'Brien, the villainous Party member in Orwell's novel, is embodied in an all-knowing computer that the author interrogates. The book summarizes the predictions made by noted authorities in 1964 about what this year would be like and grades their performances.

1984 Revisited, edited by Irving Howe (Harper and Row), is a scholarly collection of thirteen essays by various authors. The volume has as many viewpoints as authors—some essayists arguing for a revived Cold War to contain the U.S.S.R., others maintaining that the totalitarianism evident in the world today is nothing like the Ingsoc (English Socialism) of Orwell's Oceania.

One of the more ambitious Orwellian events took place the morning of January 1. A select group of artists gathered in Paris and New York for a multimedia artistic exchange via satellite TV. *Good Morning, Mr. Orwell* was arranged by video director Nam June Paik, who calls Orwell "the first media philosopher, the first communications prophet." The hour-long show, broadcast live simultaneously in both New York and Paris, featured poet Allen Ginsberg, musicians Laurie Anderson, Peter Gabriel, Oingo Boingo, Philip Glass, and John Cage, choreographer Merce Cunningham, all-around guru Salvador Dali, and performance artist Pierre-Alain Hubert. *Good Morning, Mr. Orwell* was fairly successful at demonstrating that TV and technology have not brought about Orwell's grim visions of media totalitarianism.

Despite the differences in assessment of Orwell's politics and motivations, the major themes of *Nineteen Eighty-four* are pretty clear-cut. In some ways, Orwell assumed the worst. At the same time, he incorporated much of the reality he had experienced in postwar England and had observed in Germany and the U.S.S.R. from afar.

Nineteen Eighty-four is a bitter meal, strong food for thought. If you're a glutton for punishment, you might enjoy the rich

GOTO page 181, column 1

Orwell

continued from page 179

body of other utopian and dystopian literature that's often overlooked in favor of Orwell. Some of the better examples are *The Iron Heel*, by Jack London; *The Futurological Congress*, by Stanislaw Lem; *Looking Backward*, by Edward Bellamy; *War with the Newts*, by Karel Capek; *Anthem*, by Ayn Rand; *The Lomokome Papers*, by Herman Wouk; *It Couldn't Happen Here*, by Sinclair Lewis; *Men Like Gods*, by H.G. Wells; *We*, by Evgenii Zamiatin; *Stand on Zanzibar*, *The Sheep Look Up*, and *The Jagged Orbit*, by John Brunner; *Brave New World* and *Brave New World Revisited*, by Aldous Huxley; and *1985*, by Anthony Burgess.

It was on April 4, 1984, that Winston Smith started writing a diary at the beginning of Orwell's novel. This forbidden act on Smith's part is indicative of how Orwell felt about the repressive society he envisioned. Nothing could be more horrible to imagine than a world where an individual could not write down on paper opinions or anything else personal. Smith is terrified when he sees that he's written, almost without thinking, "Down with Big Brother."

It's worth remembering, on the real April 4, 1984, that some of us (in this country, at least) are not afraid to write "Down with Big Brother" on the most prominent billboard we can find. DH

Industry

continued from page 175

Rumor: According to some people's figures, it costs $8 million to develop a monster software hit, such as *1-2-3* or *WordStar*, a monster that will pull in over $30 million at the box office.

Prediction: Lotus's Mitch Kapor, speaking in New Orleans at Softcon, had this to say about the coming shakeout in software: "Competition is forcing a shakeout, yes; but companies aren't going to go up in flames like the *Hindenburg*, or go under like the *Titanic*. They're more like the ships of Magellan's fleet. Some will make it around the globe; others will lose their way and sink almost without a sound."

Update: Coleco admitted to manufacturing only ninety-five thousand Adam home computers in 1983. Last year, the company said it would ship at least four hundred thousand Adams before the new year. In December, J.C. Penney's announced that the Adam did not meet the retailer's "quality standards" and decided to cancel catalog orders for the machines (Penney's stopped selling home computers altogether on February 1). Undaunted, Coleco has announced that it plans to introduce a line of telephones in June aimed at the teenage market.

Update: Commodore's founder Jack Tramiel resigned in mid-January and the company chose Marshall F. Smith as its new president and chief executive officer. Smith has a background in manufacturing and corporate finance and is an old buddy of Commodore's chairman Irving Gould.

Obit: Texas Instruments was thoroughly thrashed by a $660-million loss in its discontinued home computer line and closed fiscal 1983 with an overall loss of $145.5 million on total sales of $4.58 billion. The semiconductor division of the company saved the day somewhat.

Obit: Mattel, battered by losses in its video game and home computer businesses, is selling off all its divisions and planning to concentrate on its profitable toy and hobby departments. Mattel Electronics (now Intellivision Incorporated), which lost $283.5 million in the first three months of fiscal 1983, was purchased by Mattel executive Terrence E. Valeski and two backers, Ike Perlmutter and Bernard Marden—who together own New York-based Odd Lot Trading, a retail firm specializing in closeout merchandise. Mattel has also bid farewell to its Monogram Models and Western Publishing subsidiaries.

More obits: Victor Technologies filed for reorganization under Chapter 11 of the federal bankruptcy code, even though its Victor 9000 personal computer had won praise for its technological capabilities. The company owes $110 million. Timex, producer of the popular Timex/Sinclair 1000 home computer, announced that it will halt the sale of home computers. Two units introduced by the firm last year did not sell well.

Acquisitions: Cupertino-based data processing firm Tymshare has been acquired by McDonnell Douglas for $307.5 million—12.3 million outstanding shares at $25 each. Disk drive manufacturer Xebec is acquiring Evotek, a company that specializes in thin-film-based Winchester technology.

Rumor: The Japanese are coming.

Conclusions: Casey struck out; it took Hank Aaron twenty years to beat Babe Ruth's home run record; and Billy Martin is alive and well and not managing the New York Yankees.

Final conclusion: Some players will strike out and some will hit home runs, but no player owns the baseballs. DH

integrated software
integrated software
integrated software
integrated software
integrated software
integrated software
integrated software
integrated software

It's not all the same, you know.

Everybody's got integrated software these days.

But most everybody else's "integrated" software is a collection of separate programs that happen to be on the same disk. Or don't do the job you expect them to.

Something you won't find with **The New Context MBA**™ and **The Corporate MBA**™.

Of course, both have all the functions you'd expect of an integrated software product: Spreadsheet, Database Management, Business Graphics, Word Processing, Telecommunications and Forms Creation.

They're fast, too. MS-DOS™ compatibility means each of the functions work together, almost as quickly as you can define your problem.

We've got full 3270 emulation, so you don't wait in line to get data from your company's IBM mainframe.

Our new Data Exchange Facility™ lets you crank in data you've created with other programs like 1-2-3™, VisiCalc™, WordStar™, etc. No problem.

And True Data Integration™ means the changes you make in the Spreadsheet Mode get made in the Business Graphics Mode, too.

And our new Macro feature will memorize your keystrokes. So now your best work gets easier and faster the second time around.

Call us at **1-800-437-1513** (In CA, call 1-800 592-2527) for more info on The New Context MBA and The Corporate MBA.

Integrated software with a difference.

The New Context MBA.

YES. Tell me more about The New Context MBA and The Corporate MBA. 248

☐ Have a Context representative call me.

Name ______________________

Title ____________ Company ____________

Address ______________________

City ____________ State ________ Zip ________

Area Code ________ Phone Number ______________

SEND TO:

Context Management Systems, 23868 Hawthorne Blvd., Torrance, CA 90505.

Context

MANAGEMENT SYSTEMS

Personal Computer Software For Business Decisions.

NEWSBITS

Δ **DARPA Chips In.** College students studying design and architecture of integrated-circuit chips—particularly VLSI (very large scale integration) projects—will be able to have chips they design manufactured free of charge at the Defense Advanced Research Projects Agency (DARPA) VLSI fabrication center. The program is restricted to U.S. universities, and applications to use the facility are made through the National Science Foundation in Washington, D.C. The idea is to accelerate the "professional learning cycle" by giving undergraduates and graduate students access to the type of state-of-the-art facility they would not ordinarily encounter outside of private industry. Reportedly, once chip-design instructions are relayed to the DARPA center on one of three telecommunications networks, it usually takes no more than three or four weeks to deliver the completed chips.

Δ **Intellivision 3-D—Just When You Thought It Was Safe To Play Video Games Again.** A research engineer at the Georgia Institute of Technology in Atlanta, Richard Steenblik, has developed a new process for conveying three-dimensional depth to computer-generated images. In January, Georgia Tech applied for a patent on Steenblik's process. Although the details are still secret, the process is apparently simple in design but still requires that viewers wear special glasses similar to those needed for 3-D movies. The difference, reportedly, between Steenblik's 3-D scheme and the more familiar anaglyph process—wherein stereo vision is achieved by superimposition of a separate image for each eye—is that only one image is displayed. A toned-down version of the process will be unveiled later this year when Intellivision (formerly Mattel Electronics)—the first licensee—begins marketing a 3-D video game for its Intellivision unit. In the game *Hover Force 3-D*, a single, maximum-discernible depth is preset by the glasses' fixed lenses. More sophisticated versions of Steenblik's process will use glasses that permit viewers literally to dial the degree of depth they desire. Steenblik says his process will work with just about all current computer graphics equipment and is capable of both color and black-and-white graphics.

Δ **Wookie Bait.** Ropet is a personal robot that does not require an external computer for operation. Marketed by Personal Robotics Corporation, based in San Jose, California, Ropet uses a Z-80 processor and S-100 bus. Plug-in cartridges provide programs for security, entertainment, and education. External communications are available for connecting Ropet to a development station. Ropet is mobile, can avoid collisions, and obeys spoken commands.

Δ **Heavy-Metal Motor City Bash.** The NCC/COMDEX/CES of robotics shows, Robots 8, will be held June 4–7 at Cobb Hall in Detroit, Michigan. Sponsored by the Robot Institute of America and Robotics International of the Society of Manufacturing Engineers, Robots 8 should attract some two hundred different industrial robots, demonstrating applications for assembly, finishing, painting, welding, machine loading, material handling, quality control, and break dancing (just kidding). For more information, contact the Robotics Institute of America in Dearborn, Michigan. ▲

Editor **David Hunter**

Contributors **Michael Ferris, Judith Pfeffer, Marsha Stewart**

COLORPLUS™
Shatters The Mold.

A vibrant, varied new world of color is now at your fingertips.

We've shattered the mold for color graphics adapters with COLORPLUS, a high resolution, full color breakthrough fully compatible with IBM Personal Computer hardware and software.

COLORPLUS produces incredibly sharp, detailed multicolor graphics of professional quality at a low PC price.* Its state of the art biplanar technology enables high resolution 4 color 80 character graphics or medium resolution 16 color 40 character graphics.

The current IBM PC capabilities of high resolution 1 color 80 character graphics and medium resolution 4 color 40 character graphics simply pale in comparison, now that the era of COLORPLUS has arrived.

Using only a single expansion slot, COLORPLUS includes an integrated parallel port, thereby saving you an expansion slot for future needs. No modification is required to existing software in the IBM modes. Enhanced software supporting COLORPLUS Graphics capabilities is currently available from major software suppliers.

For over 31 years, Frederick Electronics Corporation has developed high quality electronic equipment world-wide. The key to our success is our international reputation for customer support. PC+ Products will continue in this tradition. Frederick Electronics Corporation is a subsidiary of Plantronics, Inc., a publicly owned company listed on the NYSE as PLX.

The goal of PC+ Products is to expand your options, to make your personal computer more valuable and productive. COLORPLUS is just one example of our commitment to that objective.

Contact us for the name of your nearest COLORPLUS dealer.

PC+™ PRODUCTS

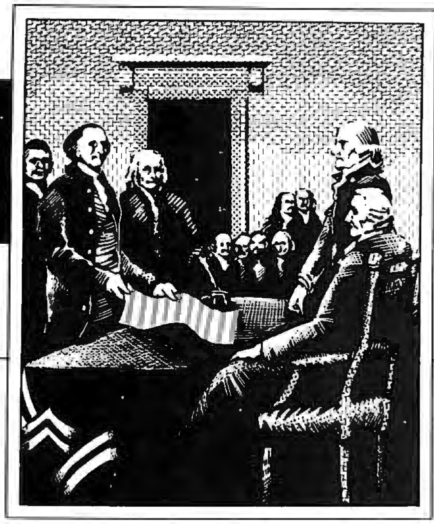
PC+ Products
Plantronics Enhanced Graphic Products, Inc.

1751 McCarthy Boulevard
Milpitas, CA 95035

408 945-8711   800 638-6211

TELEX: 17-1139

PC+ and COLORPLUS are trademarks of Plantronics/Frederick Electronics Corp.   IBM is a registered trademark of International Business Machines Corp.



THE RIGHT TO ASSEMBLE

by Ray Duncan

The LIST Utility

Although PC-DOS includes a *print* command that you can use to spool previously formatted text files to the printer, there's no simple command that properly paginates a free-form text file, such as an assembly language source file.

Therefore, this month's column presents a small utility called LIST, which you can use to print almost any type of text file—complete with page numbers and an optional title. LIST strips the high bit off all characters and discards oddball control codes, so it can also be used to get a "raw" listing of *WordStar* documents and other specially formatted word processing files. Tabs are expanded, and embedded form feed codes are recognized and properly handled.

LIST requires DOS or MS-DOS 2.0. Its output to the printer is directed through the predefined "handle" for the "standard printer device." Full support is included for pathnames in the input file specification.

How To Use LIST

The command line format for LIST is:

A⟩LIST drive:path\filename.ext ["title text"] [/C]

The drive and path, if omitted, default to the "current" or default disk drive and the current subdirectory. The parameters in square brackets are optional. The "title text" can be anything up to sixty characters long; if supplied, it's printed in the upper left corner of each page. If you have an Epson or IBM dot-matrix printer, you can use the /C switch to turn on "compressed printing"; in this mode lines up to 128 characters long will fit on a normal page.

For example, to list the file Myfile.doc on the printer in compressed mode with the title line "A Little Listing", you would enter:

A⟩LIST MYFILE.DOC "A Little Listing" /C

Three possible error messages may be received as a result of your running LIST. "Requires PC-DOS version 2.0 or greater" means that you have tried to run the program under DOS 1.0 or 1.1; no support for handles or extended file and record functions was included in that version of the operating system. "Missing file name" means you forgot to put a file specification in the command line or you put it in the wrong position (such as after the title text or switch). And "Cannot find input file" means the file doesn't exist in the current or specified subdirectory or the specified path doesn't exist. If you specify an illegal switch in the command line, LIST will simply ignore it and print the file in the uncompressed font.

A Look at LIST Internals

LIST is written in a highly modular fashion and contains many small, independent subroutines that you'll find useful in your other programming efforts. It's written to be assembled into an EXE file, so it contains three program segments: an executable code segment named CSEG, which is defined by SEGMENT and ENDS commands in lines 38 and 382; a data segment containing variables and constants, in lines 385 through 460; and a stack segment used as a scratch area by PUSHes, POPs, and CALLs, in lines 463 through 468.

The top, or control, procedure of the program is named LIST and is defined in lines 43 through 157. This procedure contains the logic of the program and makes all the decisions, implementing them through successive calls on the smaller subroutines found in lines 160 through 380.

The procedure GET_FILENAME extracts the path and file specification of the desired input file from the command line tail in the program segment prefix and transfers them into a local working buffer for use by the OPEN_INPUT routine. It returns the carry flag set or cleared to let the calling routine know whether a legal filename was found.

Similarly, GET_TITLE fetches the title text delimited by two quote marks from the command line tail into the page heading string in HEADING_BUFFER. GET_SWITCH scans for the /C switch in the command tail and sets the variable COMPRESS_SWITCH to indicate the presence or absence of that switch. Since the title and compress switches are both optional, these routines take no error action if they don't find what they're looking for.

OPEN_INPUT uses the path and file specification left in the local buffer by GET_FILENAME to try to open the input file for reading only, using the DOS 2.0 "extended file function" 3DH. The "handle" or token for the file that is returned by DOS is saved in the variable INPUT_HANDLE and used in future read-record calls. CLOSE_INPUT is called at the end of the program to close the same file with function 3EH before LIST exits; this is not strictly necessary, since the file was used only for reading, but it's included here for demonstration purposes.

GET_CHAR fetches one character from the input file deblocking buffer. If the buffer has been used up, it calls READ_BLOCK to get another chunk of data from the disk. It maintains a pointer, called INPUT_PTR, to describe its position in the deblocking buffer. PUT_CHAR appends one character to the string that's being built up in the output buffer; it maintains the buffer position indicator OUTPUT_PTR.

READ_BLOCK reads a record—of length defined by the equate BLKSIZE—from the input file. It initializes the pointer INPUT_PTR for use by GET_CHAR. If the file doesn't contain an end-of-file mark (01AH or ∧Z), READ_BLOCK supplies one when it encounters the physical end of file.

WRITE_LINE appends a line feed to the forming output string and unilaterally sends the contents of the output buffer to the list device. WRITE_MAYBE checks to see if anything is in the buffer, by examining OUTPUT_PTR; it calls WRITE_LINE only if some characters are waiting to be printed.

PRINT_HEADING maintains and formats the current page number, sends the contents of the heading buffer to the list device, and then initializes the line count for the current page. A particularly cute and useful code sequence in this routine can be found in lines 339 through 341:

```
MOV  AX,PAGECOUNT
AAM
ADD  AX,'00'
```

This sequence loads a binary page number from a variable into AX and uses the rather obscure "Adjust Result of ASCII Multiplication" command to turn it into two ASCII digits in only three instructions! Unfortunately, it can't handle page numbers larger than 99. A more general-purpose conversion routine will be presented and discussed next month.

HEADING_MAYBE checks the line count to see if the current page is full and calls PRINT_HEADING as necessary to go to the top of a new page and print the contents of the heading buffer. INIT_BUFF is called at the beginning of the program to initialize the input file deblocking buffer and pointer and the output buffer pointer.

Finally, COMPRESS_ON and COMPRESS_OFF use the command strings defined in the data segment to enable and disable, respectively, compressed printing.

Customizing LIST

This program makes extensive use of equates, so you can easily modify it to suit other printers or listing formats. Let's look at a few of the more interesting of these equates.

Blksize, in line 31, sets the size of the deblocking buffer. In the supplied listing, it is set to 1024, which means that data is read in from the input file in 1024-byte chunks. If you have a large external printer buffer or a very fast line printer, you might wish to increase the size of this deblocking buffer so that the data can be transferred to the list device with fewer disk accesses.

Linesize, in line 32, sets the maximum line length, in characters, that the printer will accept; it is supplied as 132, which is appropriate for an Epson printer in compressed mode. Buffers and subroutines throughout the program reference this equate. If a line of text exceeding Linesize is read from the file, it will be broken into pieces and printed on separate lines.

Pagesize, in line 33, specifies the number of lines, including the heading, that will be printed on a single page. For normal nine-by-eleven continuous-form paper, values of 55 to 58 are appropriate.

Heading_lines, in line 34 of the program listing, defines the total number of lines to be printed in the page heading. The first line of the heading always contains the title and page number, and the remainder of the lines are blank. The value of this equate, as supplied, is 3.

Output_handle, in line 36, defines the predefined handle, or token, that is used by text-output subroutines to write to the list device. In the supplied program, Output_handle is equated to 4, which is specified on page D-15 of the DOS 2.0 manual as the "Standard Printer Device." By changing this value to 1, which is the "Standard Output Device," you may redirect the output of LIST to the serial port, console, or a disk file—as well as to the line printer.

You can easily expand or modify the format of the page heading by editing the data statements from heading_buffer (line 531) up to, but not including, heading_length. Leave heading_length alone; the print_heading subroutine references this equate to find the length of the heading's text string.

If you're not using an Epson or IBM printer, you can modify the strings comp_command (line 406) and norm_command (line 409), substituting the proper control codes for your printer. The control strings can be of any length; the equates comp_command_length and norm_command_length will be adjusted automatically by the assembler, and the compress_on and compress_off subroutines will compensate accordingly.

Enhancements to LIST

There are many improvements to this program that the industrious reader might wish to add. Among them are time- and date-stamping of the listing, the ability to accept multiple file names, and support for other printing modes, such as the italic or emphasized fonts. We'll discuss some of these topics next month.

Don't forget that if you're a subscriber to CompuServe, you can download the source file for this program from the XA4 database in the IBM PC Special Interest Group area. ➤

```
                                name    list
                                page    55,132
                                title   'LIST — Print text file'

                        ; LIST — a utility to print out text files with titles and page
                        ; numbers on the current list device. The high bit of all
                        ; characters is stripped, so that raw WordStar or other word
                        ; processing files can be listed. Embedded form feed codes are
                        ; recognized, and tabs are expanded. Unknown control codes (such
                        ; as the ^B and ^U codes found in WordStar documents) are discarded.
                        ; If using an Epson printer, compressed mode can be turned on with
                        ; /C switch in command so that lines up to 128 characters long will
                        ; fit on a normal page. Requires PC-DOS 2.0 or MS-DOS 2.0.

                        ; Used in the form:
                        ; A>LIST path\filename.ext ["title text"] [/C]
                        ; (items in square brackets are optional)

                        ; version 1.0   February 28, 1984
                        ; Copyright (c) 1984 by Ray Duncan
                        ; May be freely reproduced for noncommercial use.

 = 000D                 cr      equ     0dh             ;ASCII carriage return
 = 000A                 lf      equ     0ah             ;ASCII line feed
 = 000C                 ff      equ     0ch             ;ASCII form feed
 = 001A                 eof     equ     01ah            ;End of file marker
 = 0009                 tab     equ     09h             ;ASCII tab character

 = 0080                 command equ     80h             ;buffer for command tail

 = 0400                 blksize equ     1024            ;size of block reads from input file
 = 0084                 linesize equ    132             ;maximum length of output line
 = 003A                 pagesize equ    58              ;number of lines per page
 = 0003                 heading_lines equ 3             ;number of lines in page heading

 = 0004                 output_handle equ 4             ;handle of standard list device

 0000                   cseg    segment para public 'CODE'

                                assume  cs:cseg,ds:data,es:data,ss:stack

 0000                   list    proc    far             ;entry point from PC-DOS

 0000   1E                      push    ds              ;save DS:0000 for final
 0001   33 C0                   xor     ax,ax           ;return to PC-DOS
 0003   50                      push    ax
 0004   B8 ---- R               mov     ax,data         ;make our data segment
 0007   8E C0                   mov     es,ax           ;addressable via ES register.
 0009   E8 00F4 R               call    get_title       ;get listing title from
                                                        ;command line tail.
 000C   E8 0116 R               call    get_switch      ;look for /C switch if any
 000F   E8 00C5 R               call    get_filename    ;get path and file spec. for
                                                        ;input file from command line tail.
 0012   8C C0                   mov     ax,es           ;set DS=ES for remainder
 0014   8E D8                   mov     ds,ax           ;of program.
 0016   73 06                   jnc     list13          ;jump,got acceptable name.
 0018   BA 006C R               mov     dx,offset msg2  ;missing or illegal filespec,
 001B   E9 00C0 R               jmp     list9           ;print error message and exit.

 001E                   list13:                         ;make sure we're running under DOS 2.0.
 001E   B4 30                   mov     ah,30h
 0020   CD 21                   int     21h
 0022   3C 02                   cmp     al,2
 0024   73 06                   jae     list15          ;proceed,DOS 2.0 or greater
 0026   BA 0083 R               mov     dx,offset msg3  ;DOS 1.x — print error message
 0029   E9 00C0 R               jmp     list9

 002C   E8 0131 R       list15: call    open_input      ;now try to open input file
 002F   73 06                   jnc     list2           ;jump,opened input ok
 0031   BA 0050 R               mov     dx,offset msg1  ;open of input file failed,
 0034   E9 00C0 R               jmp     list9           ;print error msg and exit.

 0037                   list2:
 0037   F7 06 004C R FFFF       test    compress_switch,-1 ;was /C switch found?
 003D   74 03                   jz      list25          ;no,jump
 003F   E8 0200 R               call    compress_on     ;yes,turn on compressed print mode

 0042   E8 01F6 R       list25: call    init_buff       ;initialize input deblocking buffer

                                                        ;file successfully opened,
 0045                   list3:                          ;now print it!
 0045   E8 0147 R               call    get_char        ;read 1 character from input.
 0048   24 7F                   and     al,07fh         ;strip off the high bit
 004A   3C 20                   cmp     al,20h          ;is it a control code?
 004C   73 1D                   jae     list4           ;no,write it to list device.
                                                        ;yes it is control code,
 004E   3C 1A                   cmp     al,eof          ;is it end of file marker?
 0050   74 54                   je      list8           ;yes,jump to close files.
 0052   3C 09                   cmp     al,tab          ;is it a tab command?
 0054   74 26                   je      list5           ;yes,jump to special processing.
 0056   3C 0C                   cmp     al,ff           ;is it a form feed?
 0058   74 3C                   je      list6           ;yes,jump to special processing.
 005A   3C 0A                   cmp     al,lf           ;is it a line feed?
 005C   74 40                   je      list7           ;yes,jump to special processing.
 005E   3C 0D                   cmp     al,cr           ;if illegal control code,
 0060   75 E3                   jne     list3           ;discard it and get next char.
 0062   C7 06 0046 R 0000       mov     column,0        ;if carriage return,store it into output
 0068   EB 05 90                jmp     list45          ;string and initialize column count.

 006B                   list4:                          ;count chars. sent on this line.
 006B   FF 06 0046 R            inc     column

 006F                   list45:                         ;write this character into
 006F   E8 0161 R               call    put_char        ;forming output string.
                                                        ;is output buffer about to overflow?
 0072   81 3E 0044 R 0083       cmp     output_ptr,linesize-1
 0078   74 24                   je      list7           ;yes,force print of buffer.
 007A   EB C9                   jmp     list3           ;no,get next char. from input file.

 007C                   list5:                          ;process tab character
 007C   A1 0046 R               mov     ax,column       ;let DX:AX=column count
 007F   99                      cwd
 0080   B9 0008                 mov     cx,8            ;divide it by eight. . .
 0083   F7 F9                   idiv    cx
 0085   2B CA                   sub     cx,dx           ;remainder is in DX.
 0087   01 0E 0046 R            add     column,cx       ;update column pointer.
 008B                   list55:                         ;8 minus the remainder
 008B   51                      push    cx              ;gives us the number of
 008C   B0 20                   mov     al,20h          ;spaces to send out to
 008E   E8 0161 R               call    put_char        ;move to the next tab position
 0091   59                      pop     cx              ;restore space count
 0092   E2 F7                   loop    list55
 0094   EB AF                   jmp     short list3     ;get next character

 0096                   list6:                          ;form feed detected
 0096   E8 0194 R               call    write_maybe     ;if anything waiting in output
                                                        ;buffer,print it first
 0099   E8 01C8 R               call    print_heading   ;new page and print title
 009C   EB A7                   jmp     list3           ;get next character from input

 009E                   list7:                          ;line feed detected, interpreted
                                                        ;as print command.
 009E   E8 01BD R               call    heading_maybe   ;print heading if needed
 00A1   E8 019F R               call    write_line      ;print contents of text buffer
 00A4   EB 9F                   jmp     list3           ;get more from input file

 00A6                   list8:                          ;end of file detected,
 00A6   E8 0194 R               call    write_maybe     ;print anything that's waiting
                                                        ;in output buffer.
 00A9   B0 0C                   mov     al,ff           ;send form feed to finish listing.
 00AB   E8 0161 R               call    put_char
 00AE   E8 019F R               call    write_line
 00B1   E8 013E R               call    close_input     ;close input file.
                                                        ;turn off compressed print mode,
                                                        ;if it was enabled.
 00B4   F7 06 004C R FFFF       test    compress_switch,-1
 00BA   74 03                   jz      list85
 00BC   E8 020F R               call    compress_off
 00BF   CB              list85: ret                     ;now return to PC-DOS.

 00C0                   list9:                          ;come here to print error message
 00C0   B4 09                   mov     ah,9            ;and return control to PC-DOS
 00C2   CD 21                   int     21h
 00C4   CB                      ret

 00C5                   list    endp


 00C5                   get_filename proc near          ;process name of input file
                                                        ;DS:SI <- addr command line
 00C5   BE 0080                 mov     si,offset command
                                                        ;ES:DI <- addr filespec buffer
 00C8   BF 0000 R               mov     di,offset input_name
 00CB   FC                      cld
 00CC   AC                      lodsb                   ;any command line present?
 00CD   0A C0                   or      al,al           ;return error status if not.
 00CF   74 21                   jz      get_filename4
 00D1                   get_filename1:                  ;scan over leading blanks
 00D1   AC                      lodsb   ;to file name
 00D2   3C 0D                   cmp     al,cr           ;if we hit carriage return
 00D4   74 1C                   je      get_filename4
 00D6   3C 2F                   cmp     al,'/'          ;or switch,
 00D8   74 18                   je      get_filename4
 00DA   3C 22                   cmp     al,'"'          ;or quote mark,filename missing.
 00DC   74 14                   je      get_filename4   ;so go return error flag.
 00DE   3C 20                   cmp     al,20h          ;is this a blank?
 00E0   74 EF                   jz      get_filename1   ;if so keep scanning.
 00E2                   get_filename2:                  ;found first char of name,
 00E2   AA                      stosb                   ;move last char. to output
                                                        ;filename buffer.
 00E3   AC                      lodsb                   ;check next character,found
 00E4   3C 0D                   cmp     al,cr           ;carriage return yet?
 00E6   74 08                   je      get_filename3   ;yes,exit with success code.
 00E8   3C 22                   cmp     al,'"'          ;same if quote encountered.
 00EA   74 04                   je      get_filename3
 00EC   3C 20                   cmp     al,20h          ;is this a blank?
 00EE   75 F2                   jne     get_filename2   ;if not keep moving chars.
 00F0                   get_filename3:                  ;exit with carry =0
 00F0   F8                      clc                     ;for success flag
 00F1   C3                      ret
 00F2                   get_filename4:                  ;exit with carry =1
 00F2   F9                      stc                     ;for error flag
 00F3   C3                      ret
 00F4                   get_filename endp


 00F4                   get_title proc near             ;process title for listing
                                                        ;DS:SI <- addr command line
 00F4   BE 0080                 mov     si,offset command
                                                        ;ES:DI <- addr page heading buffer
 00F7   BF 0532 R               mov     di,offset heading1
 00FA   FC                      cld
 00FB   AC                      lodsb                   ;any command line present?
 00FC   0A C0                   or      al,al           ;no,exit
 00FE   74 15                   jz      get_title3
 0100                   get_title1:                     ;scan for leading (") to find title.
 0100   AC                      lodsb
 0101   3C 0D                   cmp     al,cr           ;if we hit carriage return,
 0103   74 10                   je      get_title3      ;title text is missing.
 0105   3C 22                   cmp     al,'"'          ;found delimiter?
 0107   75 F7                   jne     get_title1      ;if so keep scanning.
 0109                   get_title2:                     ;get next char. of title.
 0109   AC                      lodsb
 010A   3C 22                   cmp     al,'"'          ;terminate if 2nd (") delimiter
 010C   74 07                   je      get_title3      ;or carriage return found
 010E   3C 0D                   cmp     al,cr
 0110   74 03                   je      get_title3
 0112   AA                      stosb                   ;store this char. into page heading buffer
 0113   EB F4                   jmp     get_title2      ;examine next char.
 0115                   get_title3:
 0115   C3                      ret
 0116                   get_title endp

 0116                   get_switch proc near            ;Scan the input line for a "/"
                                                        ;delimiting a switch,then make
                                                        ;sure it is legal "/C" or "/c".
                                                        ;if legal switch found,set
                                                        ;variable COMPRESS_SWITCH true.
 0116   BE 0081                 mov     si,offset command+1 ; DS:SI = addr of command line
 0119                   get_switch1:                    ;look for "/" character
 0119   AC                      lodsb
 011A   3C 0D                   cmp     al,cr           ;if we run into a carriage return,
 011C   74 12                   jz      get_switch2     ;switch missing so take normal exit.
 011E   3C 2F                   cmp     al,'/'
 0120   75 F7                   jne     get_switch1     ;not '/' yet,keep looking.
```

```
0122  AC                    lodsb                          ;found '/',pick up next char.
0123  0C 20                 or      al,20h                 ;and fold to lower case.
0125  3C 63                 cmp     al,'c'                 ;c=compress
0127  75 07                 jne     get_switch2            ;not c,jump
                                                           ;set compress switch
0129  26: C7 06 004C R FFFF mov     es:compress_switch,-1
0130                  get_switch2:
0130  C3                    ret                            ;exit
0131                  get_switch endp

0131                  open_input proc near                 ;open input file
                                                           ;DS:DX=addr filename
0131  BA 0000 R             mov     dx,offset input_name
0134  B0 00                 mov     al,0                   ;AL=0 for read only
0136  B4 3D                 mov     ah,3dh                 ;function 3dh=open
0138  CD 21                 int     21h                    ;handle returned in AX,
013A  A3 0040 R             mov     input_handle,ax        ;save it for later.
013D  C3                    ret                            ;CY is set if error
013E                  open_input endp

013E                  close_input proc near                ;close input file
013E  8B 1E 0040 R          mov     bx,input_handle        ;BX=handle
0142  B4 3E                 mov     ah,3eh
0144  CD 21                 int     21h
0146  C3                    ret
0147                  close_input endp

0147                  get_char proc near                   ;get one character from input buffer
0147  8B 1E 0042 R          mov     bx,input_ptr           ;is pointer at end of buffer?
014B  81 FB 0400            cmp     bx,blksize
014F  75 06                 jne     get_char1              ;no,jump
                                                           ;yes,buffer is exhausted,
0151  E8 016E R             call    read_block             ;new block must be read from disk.
0154  BB 0000               mov     bx,0                   ;initialize buffer pointer.
0157                  get_char1:
0157  8A 87 00AD R          mov     al,[input_buffer+bx]
015B  43                    inc     bx                     ;bump input buffer pointer
015C  89 1E 0042 R          mov     input_ptr,bx
0160  C3                    ret
0161                  get_char endp

0161                  put_char proc near                   ;put one character into output buffer
0161  8B 1E 0044 R          mov     bx,output_ptr
0165  88 87 04AD R          mov     [output_buffer+bx],al
0169  FF 06 0044 R          inc     output_ptr             ;bump pointer to output string
016D  C3                    ret
016E                  put_char endp

016E                  read_block proc near                 ;read block of data from input file.
016E  8B 1E 0040 R          mov     bx,input_handle
0172  B9 0400               mov     cx,blksize
0175  BA 00AD R             mov     dx,offset input_buffer
0178  B4 3F                 mov     ah,3fh
017A  CD 21                 int     21h
017C  73 03                 jnc     read_block1            ;jump if no error status
017E  B8 0000               mov     ax,0                   ;simulate a zero length read if error
0181                  read_block1:
0181  3D 0400               cmp     ax,blksize             ;was full buffer read in?
0184  74 07                 je      read_block2            ;yes,jump
0186  8B D8                 mov     bx,ax                  ;no,store End-of-File mark
0188  C6 87 00AD R 1A       mov     byte ptr [input_buffer+bx],eof
018D                  read_block2:
018D  C7 06 0042 R 0000     mov     input_ptr,0            ;initialize pointer to input buffer.
0193  C3                    ret
0194                  read_block endp

0194                  write_maybe proc near                ;transmit line to list device,if
                                                           ;output buffer contains anything.
0194  A1 0044 R             mov     ax,output_ptr          ;pointer is nonzero if characters
0197  0B C0                 or      ax,ax                  ;are waiting in buffer.
0199  74 03                 jz      write_maybe1           ;nothing,jump to exit
019B  E8 019F R             call    write_line             ;something there,send it to printer
019E                  write_maybe1:
019E  C3                    ret
019F                  write_maybe endp

019F                  write_line proc near                 ;transmit contents of output
                                                           ;buffer to the standard list device.
019F  B0 0A                 mov     al,lf                  ;append line feed to string.
01A1  E8 0161 R             call    put_char
01A4  8B 0E 0044 R          mov     cx,output_ptr          ;CX contains length of string
                                                           ;DS:DX=buffer address
01A8  BA 04AD R             mov     dx,offset output_buffer
01AB  BB 0004               mov     bx,output_handle;BX=handle for standard list device.
01AE  B4 40                 mov     ah,40h                 ;function 40h=write to device.
01B0  CD 21                 int     21h                    ;request service from DOS.
01B2  FF 06 0048 R          inc     linecount              ;count lines printed this page.
01B6  C7 06 0044 R 0000     mov     output_ptr,0           ;reset pointer to list device buffer
01BC  C3                    ret
01BD                  write_line endp

01BD                  heading_maybe proc near              ;print heading if the line
                                                           ;count justifies it
01BD  83 3E 0048 R 3A       cmp     linecount,pagesize
01C2  7C 03                 jl      heading_maybe2         ;jump,page not full yet
01C4  E8 01C8 R             call    print_heading          ;form feed and print title
01C7                  heading_maybe2:
01C7  C3                    ret
01C8                  heading_maybe endp

01C8                  print_heading proc near              ;print form feed,title,and page no.
01C8  FF 06 004A R          inc     pagecount              ;bump page number,
01CC  A1 004A R             mov     ax,pagecount           ;load it,
01CF  D4 0A                 aam                            ;and turn it into ASCII,
01D1  05 3030               add     ax,'00'
01D4  88 26 0573 R          mov     heading2,ah            ;then store into heading string.
01D8  A2 0574 R             mov     heading2+1,al
                                                           ;now print the heading string.
01DB  BA 0531 R             mov     dx,offset heading_buffer
01DE  B9 0048 90            mov     cx,heading_length
01E2  BB 0004               mov     bx,output_handle
01E5  B4 40                 mov     ah,40h
01E7  CD 21                 int     21h
                                                           ;initialize line count
01E9  C7 06 0048 R 0003     mov     linecount,heading_lines
01EF  C7 06 0046 R 0000     mov     column,0               ;and column counter
01F5  C3                    ret
01F6                  print_heading endp

01F6                  init_buff proc near                  ;initialize i/o buffers
01F6  E8 016E R             call    read_block             ;read 1st block of input file
01F9  C7 06 0044 R 0000     mov     output_ptr,0           ;initialize pointer to output string
01FF  C3                    ret
0200                  init_buff endp

0200                  compress_on proc near                ;turn on compressed printing mode
                                                           ;by sending command string.
0200  B9 0001 90            mov     cx,comp_command_length
0204  BB 0004               mov     bx,output_handle
0207  BA 004E R             mov     dx,offset comp_command
020A  B4 40                 mov     ah,40h
020C  CD 21                 int     21h
020E  C3                    ret
020F                  compress_on endp

020F                  compress_off proc near               ;turn off compressed printing mode
                                                           ;by sending command string.
020F  B9 0001 90            mov     cx,norm_command_length
0213  BB 0004               mov     bx,output_handle
0216  BA 004F R             mov     dx,offset norm_command
0219  B4 40                 mov     ah,40h
021B  CD 21                 int     21h
021D  C3                    ret
021E                  compress_off endp

021E                  cseg    ends


0000                  data    segment para public 'DATA'

0000     40 [         input_name      db      64 dup (0)         ;buffer for input filespec
              00
                 ]


0040  0000            input_handle    dw      0                  ;token from PC-DOS for input file.

0042  0000            input_ptr       dw      0                  ;pointer to input blocking buffer
0044  0000            output_ptr      dw      0                  ;pointer to output blocking buffer

0046  0000            column          dw      0                  ;column count for tab processing
0048  003A            linecount       dw      pagesize           ;line counter,current page.
                                                                 ;(set to "pagesize" initially to
                                                                 ;force first heading on listing)
004A  0000            pagecount       dw      0                  ;current page number

004C  0000            compress_switch dw      0                  ;set to -1 if /C switch
                                                                 ;found in command line tail.

004E  0F              comp_command db         0fh                ;command string for compressed mode
 = 0001               comp_command_length equ $-comp_command

004F 12               norm_command db         12h                ;command string for normal print
 = 0001               norm_command_length equ $-norm_command

0050  0D 0A           msg1            db      cr,lf
0052  43 61 6E 6E 6F 74               db      'Cannot find input file.'
      20 66 69 6E 64 20
      69 6E 70 75 74 20
      66 69 6C 65 2E
0069  0D 0A 24                        db      cr,lf,'$'

006C  0D 0A           msg2            db      cr,lf
006E  4D 69 73 73 69 6E               db      'Missing filename.'
      67 20 66 69 6C 65
      20 6E 61 6D 65 2E
0080  0D 0A 24                        db      cr,lf,'$'

0083  0D 0A           msg3            db      cr,lf
0085  52 65 71 75 69 72               db      'Requires PC-DOS version 2 or greater.'
      65 73 20 50 43 2D
      44 4F 53 20 76 65
      72 73 69 6F 6E 20
      32 20 6F 72 20 67
      72 65 61 74 65 72
      2E
00AA  0D 0A 24                        db      cr,lf,'$'

00AD  0400 [          input_buffer    db      blksize dup (?)    ;deblocking buffer for input file
            ??
               ]


04AD     84 [         output_buffer   db      linesize dup (?)   ;buffer used to build output
            ??
               ]

                                                                 ;lines for list device

0531  0C              heading_buffer  db      ff                 ;form feed control code
0532     3C [         heading1        db      60 dup (' ')       ;filled in with title from user
            20
               ]

056E  50 61 67 65 20                  db      'Page '
0573  30 30 0D        heading2        db      '00',cr
0576     03 [                         db      heading_lines dup (lf)
            0A
               ]

 = 0048               heading_length  equ     $-heading_buffer


0579                  data    ends


0000                  stack   segment para stack 'STACK'
0000     40 [                 db      64 dup (?)
            ??
               ]

0040                  stack   ends

                      end     list
```

softalk presents the bestsellers

It seems at times as though nothing ever changes in the IBM software market. *1-2-3* is always the leading program, and any shuffling that occurs below it is usually of a less than momentous nature.

As is usually the case, appearances are deceiving. The difference between the IBM software market and other computer software markets is that transition is much more orderly in the IBM market. IBM Personal Computer owners are significantly less inclined to embrace every new package that comes along. Instead, they rely on proven products and switch software only when the newcomer provides measurable increases in productivity.

But that isn't to imply that no changes are occurring. Catalysts for changes in the market apparently number two at the present time, each making a slow but significant impact on what software is being purchased.

The prime cause of change for PC owners is the development of true sixteen-bit software. The current sales leaders, *1-2-3* and *Microsoft Flight Simulator*, fall into that category. The genre of software that has seen the greatest flood of programs written exclusively for the PC is word processing.

The old guard that are suffering the most from the changeover are *WordStar* and *VisiCalc*.

WordStar continues to slump, both in relation to all IBM software and in relation to its direct competitors in the word processing area. This report, showing results for the month of February, shows *WordStar* slipping to fifteenth place in the Top Thirty.

The new leader among word processing programs is *PFS:Write*. Software Publishing Corporation's program represents a new generation of word processing entries—the smaller and less expensive, the better. Previously, each new word processor had attempted to find a niche with new and better features, often at higher and higher prices. *PFS:Write* is the reaction to that trend, sticking to the basics of getting words on paper and leaving the esoterica to others.

There was barely enough difference between *WordStar* and its next three challengers to measure. *MultiMate*, such a success that its publisher has changed the company name to MultiMate International, had bested *WordStar* last month but trailed this month. Following in short order were *WordPerfect* and *Word*. All four products are fully implemented, high-priced word processors; in other words, they're the exact antithesis of *PFS:Write*.

VisiCalc suffers from the same malaise that's attacking all other spreadsheet programs in the IBM market—*1-2-3*. All competing products suffered relative to *1-2-3* sales in February. *Multiplan* dropped a few notches, and *VisiCalc* and *SuperCalc* dropped off the Top Thirty.

For *VisiCalc*, part of the problem was an accounting one. Because of the lawsuit between author Software Arts and publisher VisiCorp, competing *VisiCalc* packages are now available. Together, sales of *VisiCalc* and *VisiCalc: Advanced Version* would have made the twentieth spot on the Top Thirty. Separately, neither was strong enough to make the list.

February results represent a real setback for Sorcim. Sorcim has totally redone its top-of-the-line *SuperCalc* product to take advantage of the PC's sixteen-bit architecture, and the company had been doing a yeoman's job of reestablishing the *SuperCalc* line, gaining both in market share and real sales each month. Much of that progress was reversed in February. It seems clear that any product competitive with *1-2-3* has an uphill battle ahead of it.

The second catalyst for change in the PC market is the arrival of Junior. Owners of PCjrs are definitely a lighter-hearted, more frivolous breed.

February was the first month in which Junior owners made their preferences felt in the software market. The first beneficiaries were *Ultima II* and *Lode Runner,* two very different entertainment programs.

Ultima II is a fantasy, role-playing game. Published by Sierra On-Line, it's been a favorite on every computer for which it's been translated.

Lode Runner is an arcade game of an advanced generation. Not only does the game provide multiple levels of challenge; it incorporates the means by which a player can create new levels. The provision for multiple levels is nothing new, but the capacity to implement new designs is a recent addition to the game genre. The extra depth and value that such software contains is receiving favorable response across a broad spectrum of computers.

Neither the insertion of additional sixteen-bit software targeted directly at the PC nor the presence of PC*jr* owners portends a revolutionary shakeup in the Top Thirty. The changes apparently will be slow in coming. But come they will.

A footnote to the February results was the strong showing of a pair of tax packages. *Tax Preparer* scored in the top fifty programs, and *Tax Manager* was only slightly below that. The early interest in tax software portends a strong two months in March and April, as PC owners look toward April 15. ▲

IBM-franchised retail stores representing approximately 4.91 percent of all sales of IBM and IBM-related Personal Computer products volunteered to participate in the poll.

Respondents were contacted early in March to ascertain their sales for the month of February.

The only criterion for inclusion on the list was the number of units sold; such other criteria as quality of product, profitability to the computer store, and personal preference of the individual respondents were not considered.

Respondents in March represented every geographical area of the continental United States.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only to the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus or minus 3.08 percent, which translates roughly into the theoretical possibility of a change of 3.31 points, plus or minus, in any index number.

the top thirty

This Month	Last Month	Index	
1	1	342.15	**1-2-3**, Mitch Kapor and Jonathan Sachs; Lotus Development
2	2	98.27	**Microsoft Flight Simulator**, Bruce Artwick; Microsoft
3	3	83.71	**PFS:File**, John Page and D.D. Roberts; Software Publishing Corporation
4	10	80.98	**MasterType**, Bruce Zweig; Scarborough
5	4	69.15	**dBase II**, Wayne Ratliff; Ashton-Tate
6	9	64.60	**PFS:Write**, Sam Edwards, Brad Crain, and Ed Mitchell; Software Publishing Corporation
7	21	50.95	**Basic Compiler**, Microsoft; IBM
8	12	48.22	**PFS:Report**, John Page; Software Publishing Corporation
9	14	46.40	**Home Accountant Plus**, Mike Farmer, Bob Schoenburg, Larry Grodin, and Steve Pollack; Continental Software
10	5	43.67	**PC Tutor**, Lora Meise and Rick Lane; Comprehensive Software
11	19	42.76	**Copy II Plus**; Central Point Software
12	17	39.12	**Typing Tutor**, Michael Sierchio (Dick Ainsworth and Al Baker); IBM (Microsoft)
13	27	35.48	**Cdex Training for the IBM PC**, Rohit Patel; Cdex Corporation
14	26	31.84	**ProKey**, David Rose; Rosesoft
15	13	30.02	**WordStar**; MicroPro
16	22	29.11	**Zork I**; Infocom
17	7	28.20	**Multiplan**; Microsoft
18	6	26.38	**MultiMate**; MultiMate International
19	—	25.47	**Fortran**, Microsoft; IBM
20	30	24.56	**WordPerfect**, Alan Ashton and Bruce Bastian; Satellite Software
21	—	23.65	**Ultima II**, Jay Sullivan (Lord British); Sierra On-Line
22	24	22.74	**Zork III**; Infocom
23	8	21.83	**Word**; Microsoft
	16	21.83	**Norton Utilities**, Peter Norton; Peter Norton Inc.
25	28	20.92	**General Ledger**, John Moss and Ken Debower; IBM (BPI)
	—	20.92	**Think Tank**, David Winer and John Llewellyn, Living Videotext
27	11	20.01	**Crosstalk**; Microstuf
28	—	19.10	**Lode Runner**, Doug Smith; Broderbund Software
29	—	18.19	**Micro Cookbook**, Brian Skiba; Virtual Combinatics
30	25	17.28	**Zork II**; Infocom
	—	17.28	**Deadline**; Infocom

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

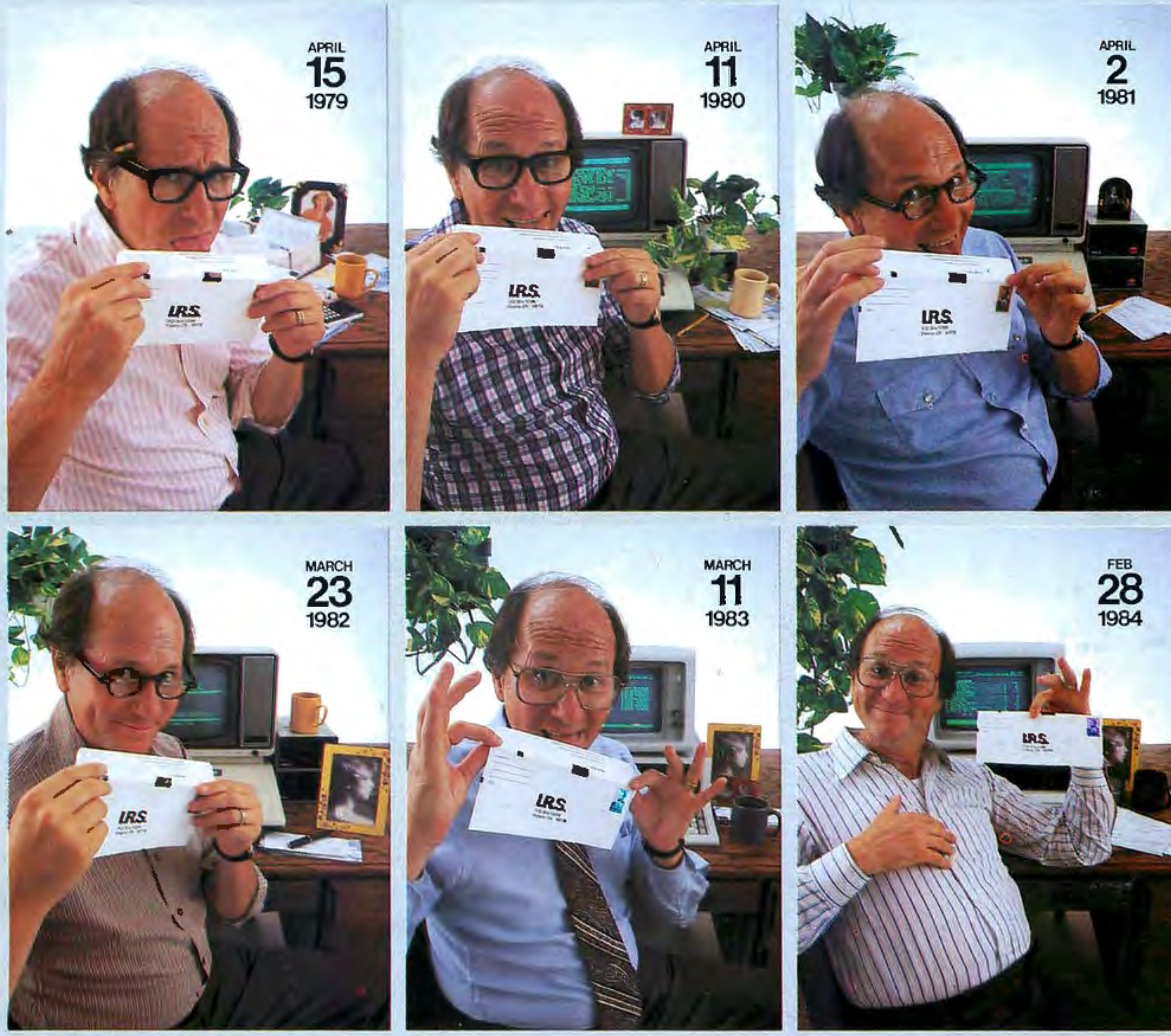
FIRST CLASS PERMIT NO. 673 NO. HOLLYWOOD, CA

POSTAGE WILL BE PAID BY ADDRESSEE

softalk

for the IBM Personal Computer

P.O. Box 7040
North Hollywood, California 91605

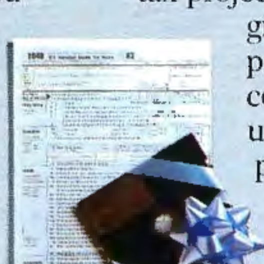


Many happy returns.

If filing a return always leaves you frazzled by forms, rattled by receipts and numbed-out by numbers, HowardSoft can help you lick this taxing situation. With HowardSoft Tax Preparer and your IBM-PC or Apple Computer, you can take the tax break you deserve.

Using HowardSoft, your tax form is right on the screen where you need it. You enter raw information just once as HowardSoft calculates quickly and accurately, makes changes automatically, then delivers error-free print-outs ready to sign and drop in the mail.

Plus, you can keep records and make tax projections year-round. HowardSoft gives you all the features of the high-priced packages at a fraction of the cost. Clear instructions, the most-used forms and schedules, and inexpensive annual updates for easy filing year after year after year.



So, why not give HowardSoft a try? Visit the computer store nearest you for a demonstration of the top-selling tax package in the country. You can count on returning happy.

Tax Preparer by HowardSoft.™
The #1 selling tax software.
8008 Girard Avenue, Suite 310, La Jolla, CA 92037 • (619) 454-0121